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.

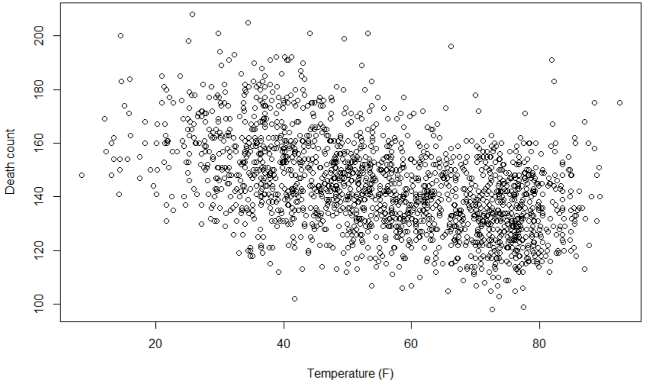
Concepts

- 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:

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

We can approximate $g(\cdot)$ using

$$y_i = \sum_{m=1}^{M} \beta_m b_m(x_i) + \epsilon_i \qquad \epsilon_i \overset{iid}{\sim} (0, \sigma^2) \ .$$

Covariate $b_m(x_i)$ may specify a linear or cubic spline.

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

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

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:

$$y_i = \beta_1 + \sum_{m=1}^{M} \beta_{1+m} b_m(x_i) \ . \tag{1}$$

Constrain the magnitude of the coefficients β_j.

Consider the ridge-regression penalty:

$$\beta_2^2 + \beta_3^2 + \ldots + \beta_{M+1}^2 \leq C, \tag{2}$$

equivalently,

$$||\boldsymbol{\beta}||_2^2 \leq C,$$

where C is an unknown positive constant.

Penalties

- Ridge regression = l2-penalty = $||\boldsymbol{\beta}||_2^2$.
- Other penalties: lasso = absolute value = l1-penalty = $||\boldsymbol{\beta}||_1 = \sum_{j=1}^{M} |\beta_j|$.
- 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 of how many knots and where to put them into a single parameter that we can choose.

Penalized Spline

Equations (1) and (2) can be written in matrix form:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} \qquad \text{with constraint } \boldsymbol{\beta}'\mathbf{B}\boldsymbol{\beta} \leq C. \tag{3}$$

Here, **B** is a diagonal matrix with 0 and 1 entries selecting which coefficients are penalized, defined below.

This problem can be equivalently formulated as

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \; (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\mathbf{B}\boldsymbol{\beta} \tag{4}$$

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

Closed-form solution

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

Differentiate wrt $\boldsymbol{\beta}$ and set to zero:

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

Closed-form solution

The least squares solution is

$$\boxed{\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X} + \lambda \mathbf{B})^{-1}\mathbf{X}'\mathbf{Y}} \tag{5}$$

for some positive number λ. Note:

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

Mortality and Temperature Example

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

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

The constraint implies a **B** matrix:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & \mathbf{0}_{1\times 40} \\ 0 & 0 & \mathbf{0}_{1\times 40} \\ \mathbf{0}_{40\times 1} & \mathbf{0}_{40\times 1} & \mathbf{I}_{40\times 40} \end{bmatrix}$$

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

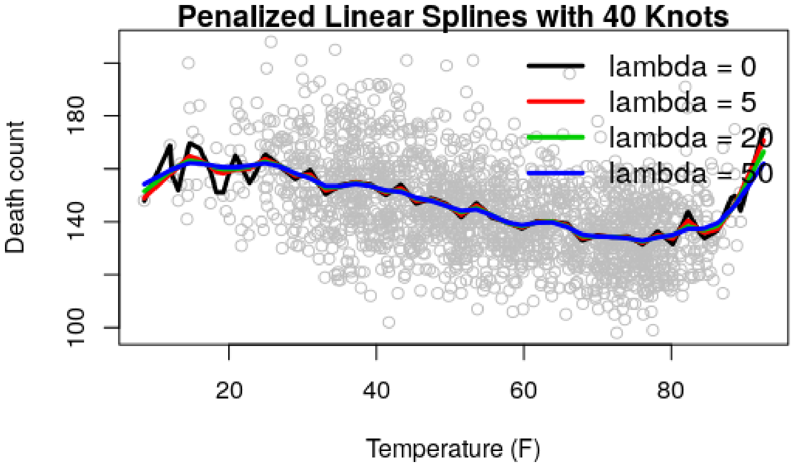
We now search through different values of λ. For each λ, we will

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

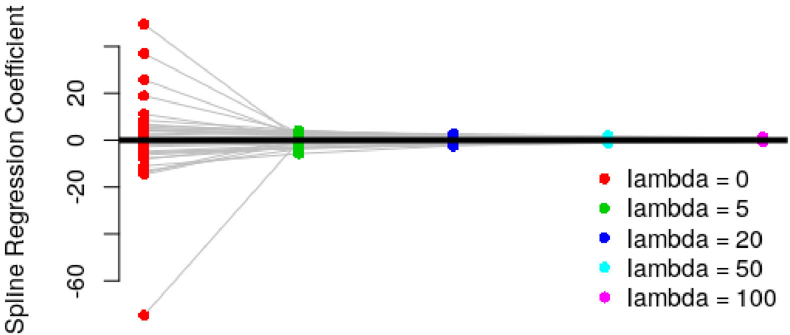
We will select the λ with the smallest GCV.

```
> 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

General principle:

- $\uparrow$ shrinkage $\rightarrow \quad \downarrow$ variance.
- $\uparrow$ shrinkage $\rightarrow \quad \uparrow$ bias.

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 $\hat{\mathbf{Y}} = \mathbf{SY}$ 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 df_{eff} as

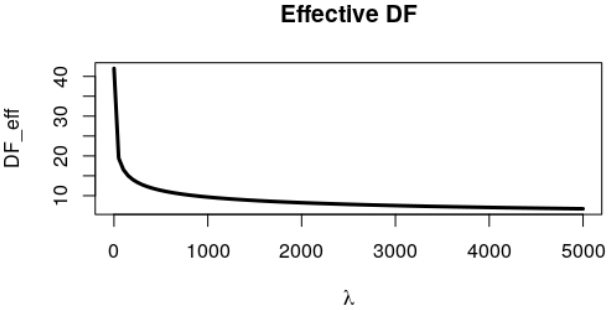
$$\boxed{df_{eff} = tr(\mathbf{S})}\,. \tag{6}$$

Note: For $\lambda > 0$, $tr(\mathbf{S}) \neq \mathrm{rank}\,\mathbf{S}$ because $\mathbf{SS} \neq \mathbf{S}$.
Hence, "effective" df.

Effective Degrees of Freedom, cont.

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

As $\lambda \uparrow$, $df_\lambda \to 2$ (since β_0 and β_1 not penalized).



Generalized Cross-validation Error, revisited

We previously defined GCV:

$$\boxed{\text{GCV} = \frac{1}{n}\,\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{[1 - n^{-1}tr(\mathbf{H})]^2}}$$

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

Now we can apply GCV to any prediction of $\mathbf{Y}$ that can be written in the form:

$$\hat{\mathbf{Y}} = \mathbf{S}\mathbf{Y}.$$

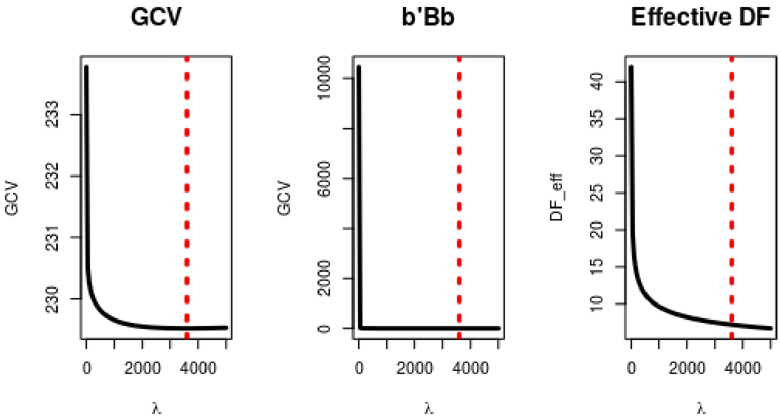
Then GCV is defined:

$$\boxed{\text{GCV} = \frac{1}{n}\,\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{[1 - n^{-1}tr(\mathbf{S})]^2}}$$

This is the definition we will use hereafter.

Smoothing Parameter Selection

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



Residual Error Variance Estimate

Recall our model is

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

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

We have two options:

$$\boxed{\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} [y_i - \hat{g}(x_i)]^2}{n - df_{\text{eff}}}}. \tag{7}$$

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

$$\boxed{\hat{\sigma}^2_{\text{unbiased}} = \frac{\sum_{i=1}^{n} [y_i - \hat{g}(x_i)]^2}{n - 2\text{tr}\{\mathbf{S}\} + \text{tr}\{\mathbf{S}\mathbf{S}'\}}}. \tag{8}$$

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 $\boldsymbol{x}_i'$ be the row vector of basis function values for x_i.

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

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

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

Confidence interval and prediction interval

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

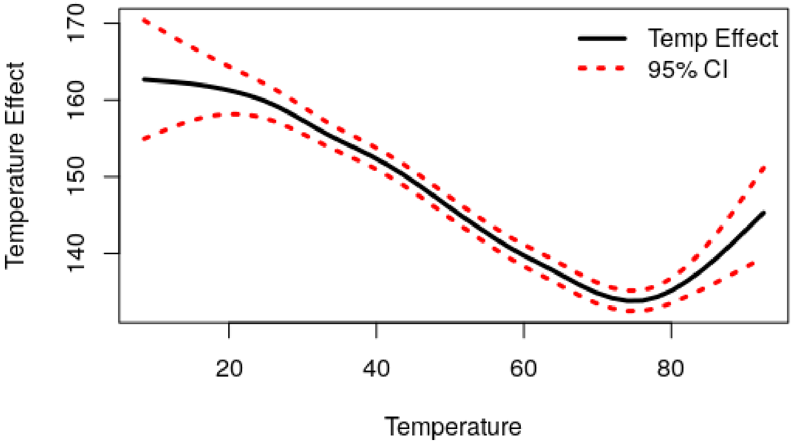
If $\lambda = 0$: the previous equation reduces to the OLS variance.

Similarly the variance for an unobserved point y_i^* with covariate x_i^* has variance

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

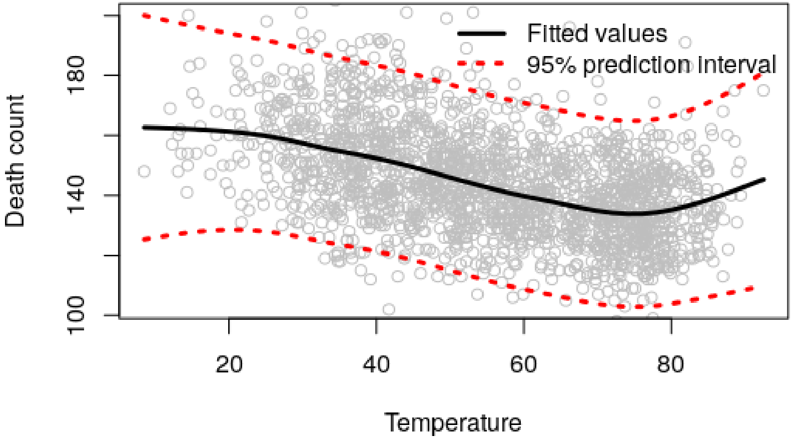
Temperature Effect on Mortality: pointwise CI

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



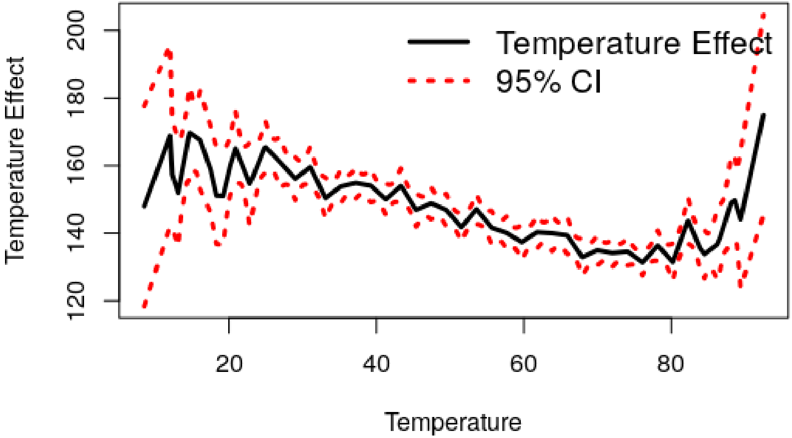
Daily Mortality Prediction

```
> 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

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:

$$\hat{g}(x) = \underset{g \in \mathcal{G}}{\arg\min} \, \{\mathbf{Y} - g(\boldsymbol{x})\}'\{\mathbf{Y} - g(\boldsymbol{x})\} + \lambda \int_a^b \{g''(x; \boldsymbol{\beta})\}^2 \, dx. \tag{9}$$

where $\mathcal{G}$ are twice-differentiable functions, $\boldsymbol{x} \in \mathbb{R}^n$ is the vector of x_i, $i = 1, \ldots, n$, and a and b is the range of x.

Smoothing spline, cont.

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

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

- 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 (9) with penalized second derivatives is equivalent to

$$(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}'\mathbf{B}\boldsymbol{\beta} \tag{10}$$

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

The key point is that (10) 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

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).
- Selection methods (default: GCV).
- Family (default: Gaussian).
- Standard error computation (default: Bayesian).

Temperature Effect on Mortality

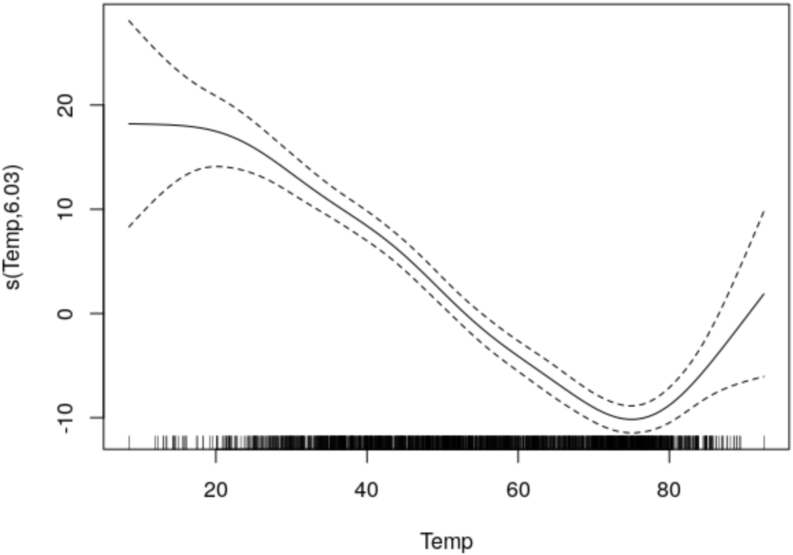
```
> library (mgcv)
> fit1 = gam(alldeaths~s(Temp), data= health)
> summary(fit1)

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
```

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
```

- 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 $\hat{g}(x_i)$.

Temperature Effect on Mortality



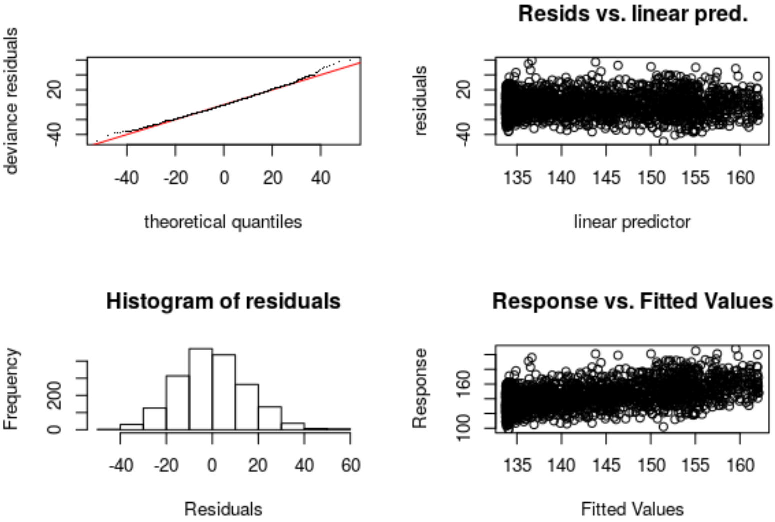
Checking gam

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

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

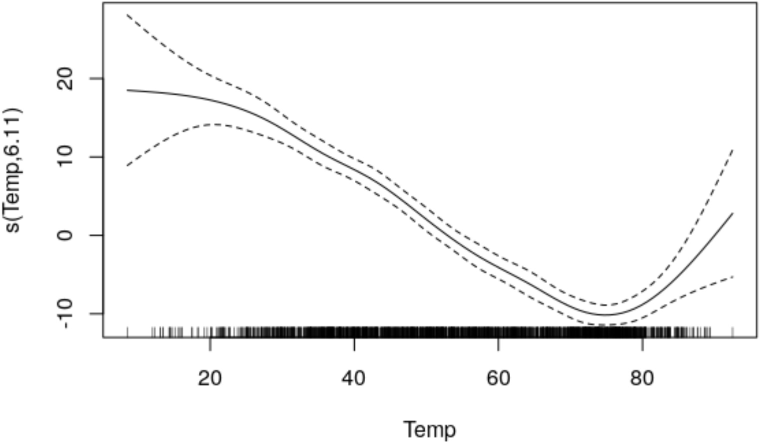
```
Method: GCV   Optimizer: magic
Smoothing parameter selection converged after 5 iterations.
The RMS GCV score gradient at convergence was 7.242e-05 .
The Hessian was positive definite.
Model rank =  10 / 10

Basis dimension (k) checking results. Low p-value (k-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
```

gam.check plots



Temperature Effect on Mortality using cubic

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



Temperature Effect on Mortality

Thin plate splines with $k = 40$.

```
> 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 ***
---
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

```
> dim (fit$Ve) ### Frequentist's
[1] 10 10
> dim (fit$Vp) ### Bayesian
[1] 10 10
```

- Fitted value

```
> fit$fitted
```

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.
- For concreteness, consider a linear spline:

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

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

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

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.

Selecting penalty using mixed models

- In mgcv::gam, we can use the option `method='REML'`
- Often results in greater smoothing

```
 > 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?

```
> # 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 relatively constant until approximately 25 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.

TO DO

add note about output and pvalues for random effects

Additive model with random intercept

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:

$$arm_{ij} = \beta_0 + g(age_{ij}) + \theta_i + \epsilon_{ij}$$

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

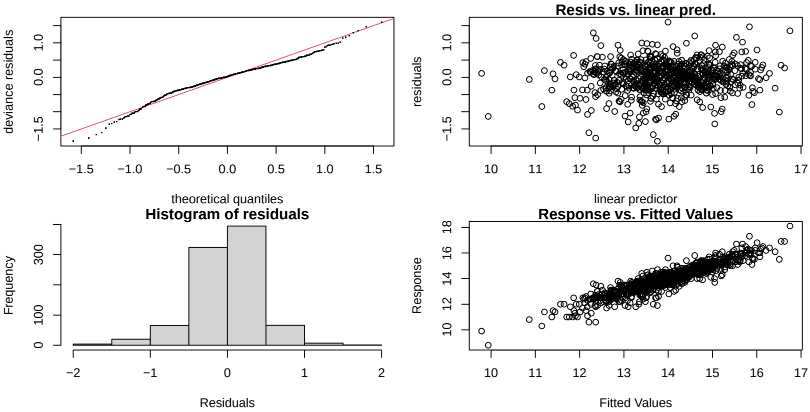
$$\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$

Additive model with random intercept

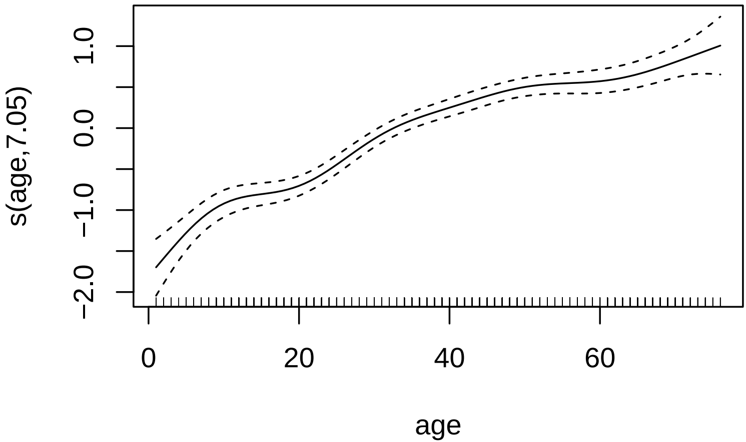
```
 fit.gamm = gam(arm~s(age)+s(id,bs = 're'),method='REML',data=nepal)
> gam.check(fit.gamm)

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-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
```

- I tend to prefer REML
- EDF somewhat close to k'. Other diagnostics okay. R code looks at $k = 20$ and results are similar, so either this model or the one with $k = 20$ is fine.

Additive mixed model with random intercept



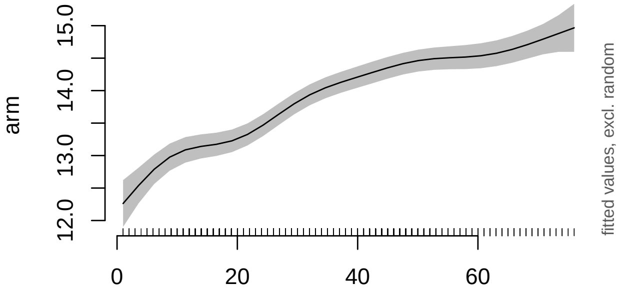
Effect of age on arm circumference



Effect of age on arm circumfenerece

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)
```