7.3 and 7.4: Basis Functions and Regression Splines

Basis Functions

- Polynomial and piece-wise constant regression models are a special case of a *basis function* approach.
- Idea: use a family of functions or transformations that can be applied to a variable X:

  $$b_1(X), b_1(X), \ldots, b_K(X)$$

  and then we fit the model

  $$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$$

- These basis functions b are fixed and known (chosen in advance).
- Like polynomial regression, these basis functions just create a new input matrix which can be used in a standard linear model.
  - So all of our linear regression inference tools apply!

Regression Splines: Piecewise Polynomials

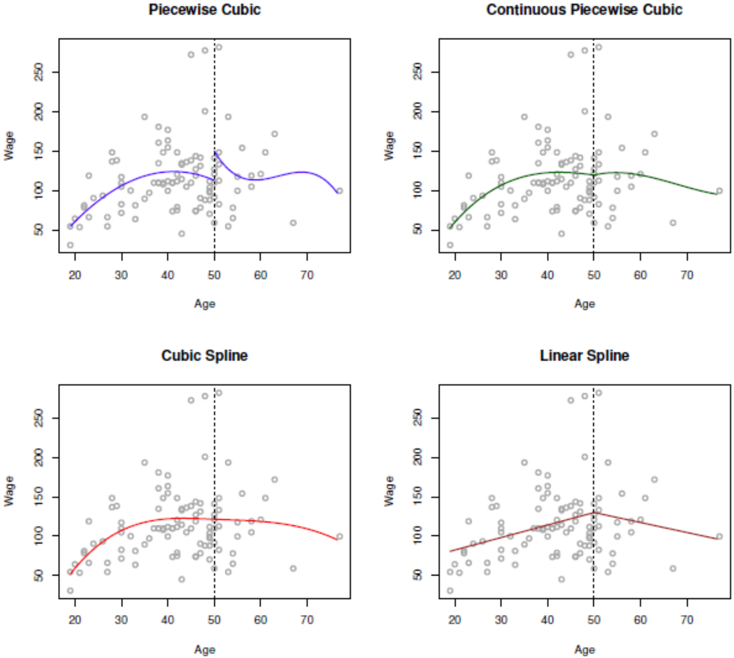
- Instead of a single polynomial in X over its whole domain, w can use different polynomials defined by knots.
- Ex:

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & \text{if} \quad x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & \text{if} \quad x_i \geq c \end{cases}$$

- Downside: piecewise approaches can create bizarre discontinuities.

Constraints and Splines

- In general, it's better to add some constraints to the polynomials.
  - One such constraint is continuity at the knots.
- *Splines* have the maximum amount of continuity.
- Each constraint *increases* degrees of freedom $+1$ by reducing the complexity of the fit.



Piecewise Cubic

Continuous Piecewise Cubic

Cubic Spline

Linear Spline

Constraints and Splines

- Top left: no constraint on the cubic polynomials.
- Top right: polynomials forced to be continuous at the knot.
    - $+1$ degree of freedom
- Bottom left: both first and second derivatives are continuous.
    - $+2$ degrees of freedom
- Bottom right: linear spline continuous at knot.

The Spline Basis Representation

- We can use the basis approach to represent regression splines.
- A linear spline with K knots can be modeled as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+1} b_{K+1}(x_i) + \epsilon_i$$

The Spline Basis Representation

We start with the basis for a linear model, then add one *truncated power basis* function per knot

$$\begin{aligned} b_1(x_i) &= x_i \\ b_{k+1}(x_i) &= (x_i - \xi_k)_+, \quad k = 1, \ldots, L \end{aligned}$$

where ξ is the knot and $(x)_+$ means the *positive part* of x:

$$(x)_+ = \begin{cases} x_i & \text{if} \quad x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Cubic Splines

A cubic spline with knots at ξ_k, $k = 1, \ldots, K$ is a piecewise cubic polynomial with continuous derivatives up to order 2 at each knot.

We can represent this as

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

where now the truncated basis function used is

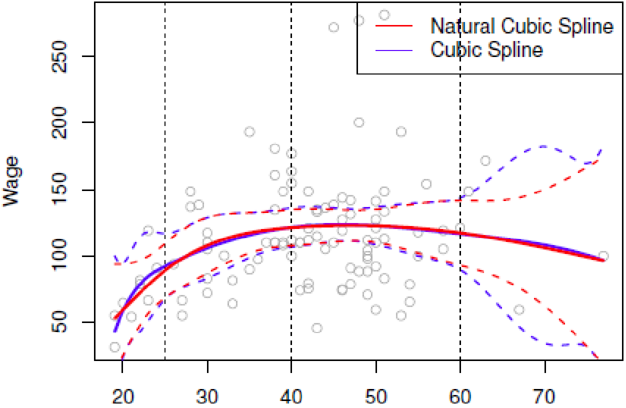
$$h(x, \xi) = (x - \xi)_+^3$$

Cubic Splines

- This amounts to fitting a model with predictors of the form $X, X^2, X^3, h(X, \xi_1), h(X, \xi_2), \ldots, h(X, \xi_K)$
  - $\xi_1, \ldots, \xi_K$ are the knots.
  - There are $K + 4$ regression coefficients (including the intercept), so the model uses $K + 4$ degrees of freedom.

Natural Cubic Splines

- Splines can have high variance at the outer range of X
- Natural splines have additional boundary constraints (linearity at boundaries)
- This adds $4 = 2 \times 2$ additional constraints.
- Estimates generally more stable at the boundaries



Choosing the Knots

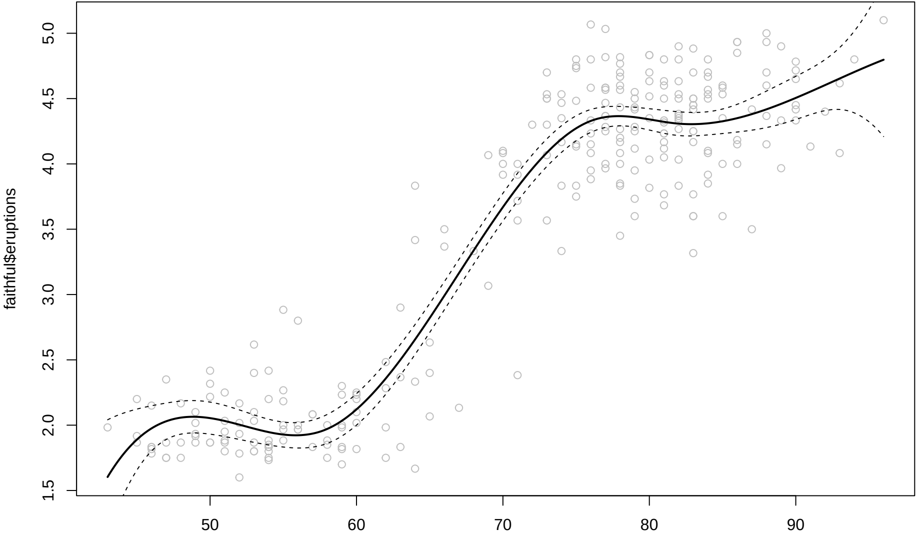
The number and location of knots needs to be set. How?

- Model most flexible in regions with many knots.
  - So we might place more knots in places where the function might vary most rapidly.
- In practice, knots are typically placed uniformly.
  - Specify desired degrees of freedom and place knots at uniform quantiles.

Regression Splines in R

```r
library(splines)
knot.q <- quantile(faithful$waiting, c(0.25,0.5,0.75))
fit <- lm(eruptions ~ bs(waiting, knots = knot.q), data = faithful)
wait.grid <- seq(43,96,0.1)
pred <- predict(fit, newdata = list(waiting = wait.grid), se = T)
plot(faithful$waiting, faithful$eruptions, col = "gray")
lines(wait.grid, pred$fit, lwd = 2)
lines(wait.grid, pred$fit + 2 * pred$se, lty = "dashed")
lines(wait.grid, pred$fit - 2 * pred$se, lty = "dashed")
```

Regression Splines in R


faithful$eruptions