

Biol. Stats II : Nonlinear models & splines

James et al., Chapter 7

Gavin Fay

02/25/2025

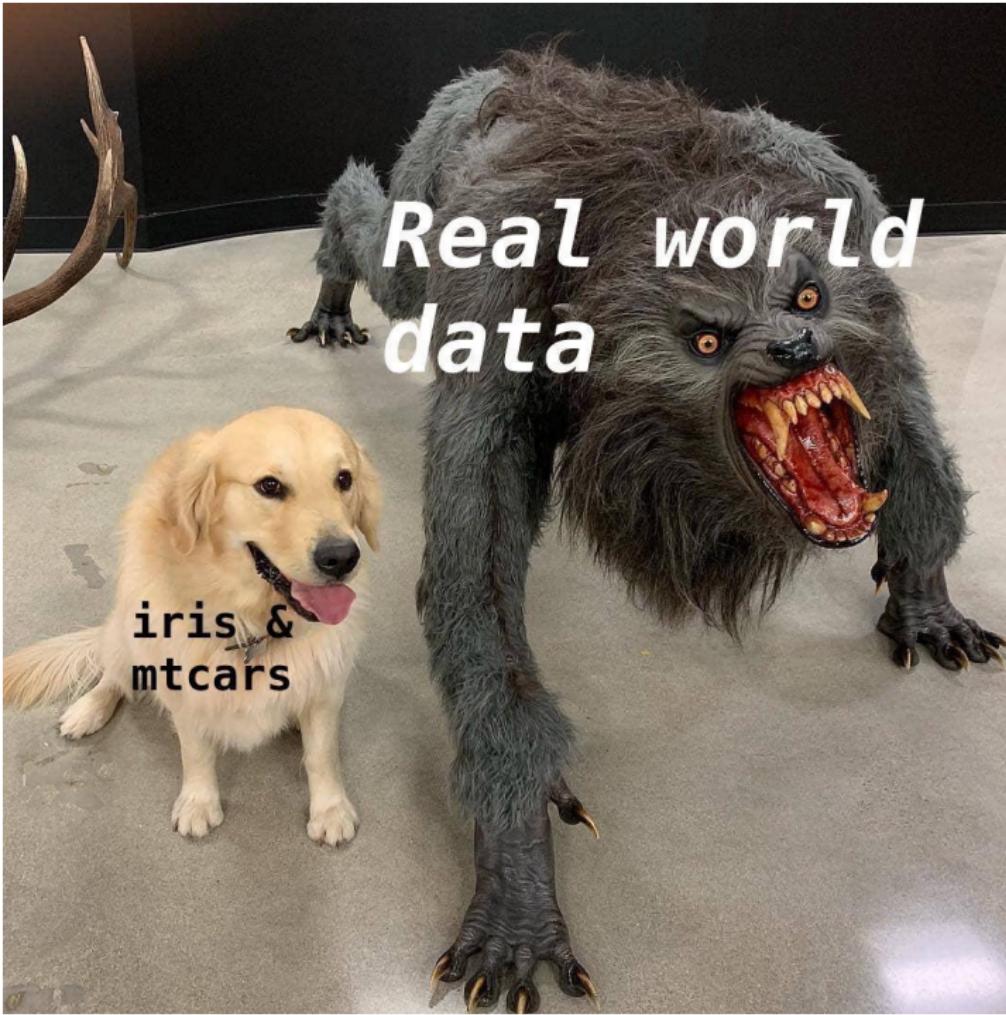
Objectives

- ▶ Discuss need for nonlinear models
- ▶ Review approaches for nonlinearity in linear models
- ▶ Polynomials
- ▶ Regression splines
- ▶ Smoothing splines

Linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

- ▶ Limitations in terms of predictive power
- ▶ Linear assumption almost always an approximation.
(sometimes a poor one)



Real world
data

iris &
mtcars

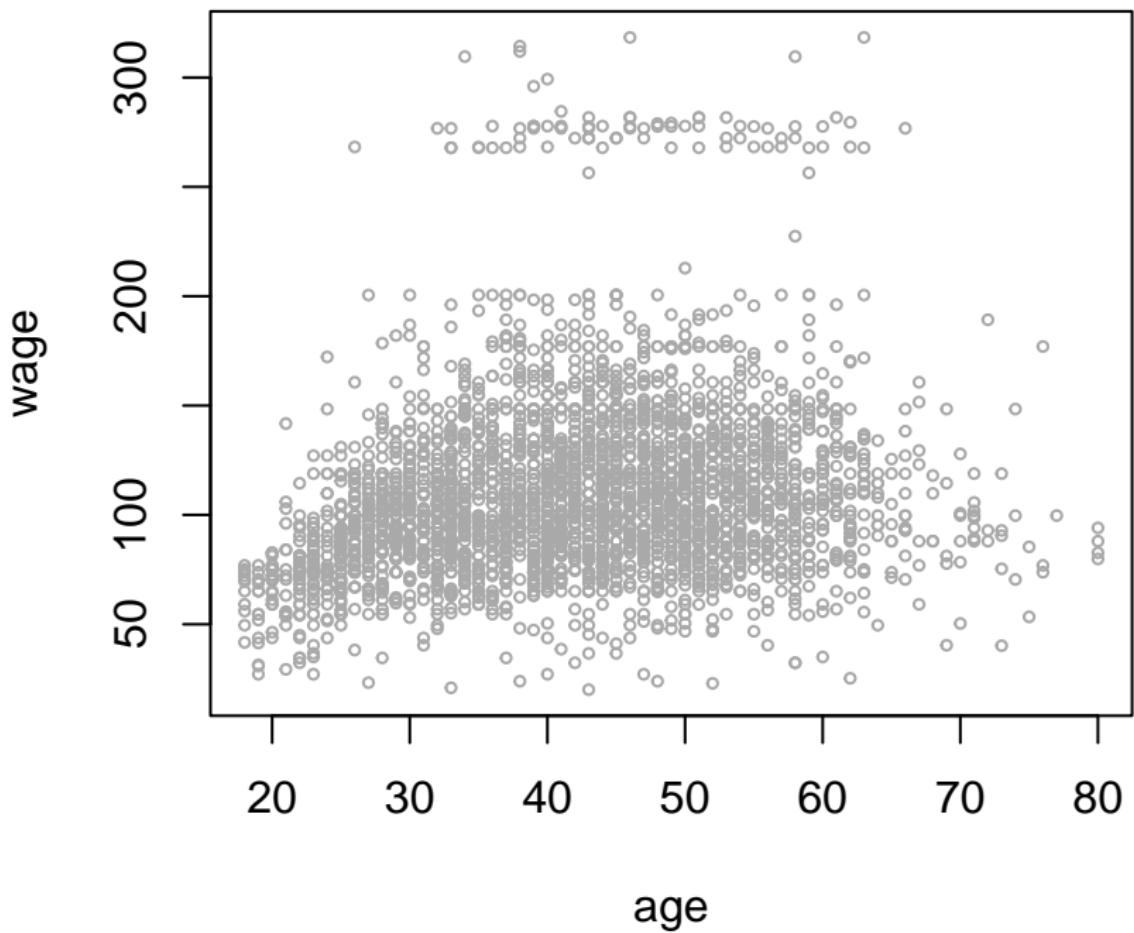
Linear regression

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

- ▶ Limitations in terms of predictive power
- ▶ Linear assumption almost always an approximation.
(sometimes a poor one)

Today's question:

- Can we relax the linear assumptions but retain interpretability?



Options for nonlinearity

Polynomial regression

- Raise original predictors to powers, e.g. x^2 , x^3 , etc.

Step functions

- Divide range of X into K regions, to create a categorical variable.
- Piecewise constant function

Regression splines

- Extension of previous two, K regions,
- Fit polynomial to the data in each region - Constraints so pieces join smoothly at region boundaries

Smoothing splines

- Minimize RSS subject to a smoothness penalty

Local regression

- Regions are allowed to overlap (e.g. LOESS)

Generalized Additive Models

- extend all of the above
- deal with multiple predictors

True nonlinear models

- The above all consider linear functions
- fit nonlinear functions of data e.g. via nls, maximum likelihood

Polynomial regression

Add additional predictors that are some degree polynomials of X.

$$y_i = \alpha + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$$

Use usual `lm()` machinery.

```
> fit=lm(wage~poly(age,4),data=Wage)
> fit2a=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
> fit2b=lm(wage~cbind(age,age^2,age^3,age^4),data=Wage)
```

```
> fit.1=lm(wage~age,data=Wage)
> fit.2=lm(wage~poly(age,2),data=Wage)
> fit.3=lm(wage~poly(age,3),data=Wage)
> fit.4=lm(wage~poly(age,4),data=Wage)
> fit.5=lm(wage~poly(age,5),data=Wage)
> anova(fit.1,fit.2,fit.3,fit.4,fit.5)
```

Analysis of Variance Table

Model 1: wage ~ age

Model 2: wage ~ poly(age, 2)

Model 3: wage ~ poly(age, 3)

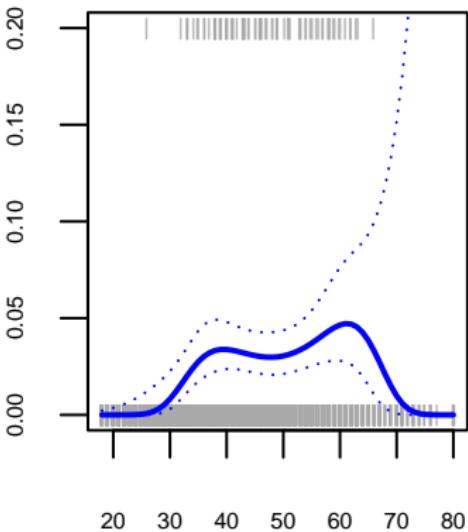
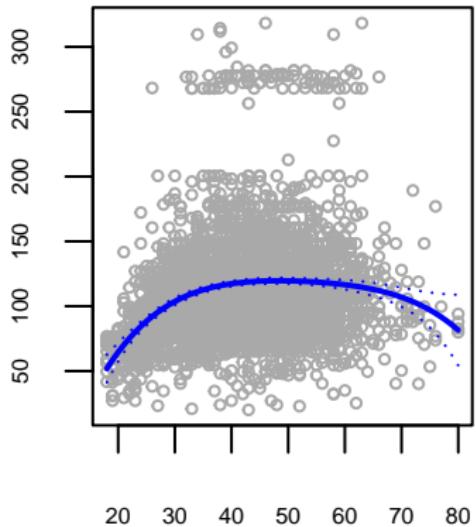
Model 4: wage ~ poly(age, 4)

Model 5: wage ~ poly(age, 5)

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	2998	5022216				
2	2997	4793430	1	228786	143.5931 <	2.2e-16 ***
3	2996	4777674	1	15756	9.8888	0.001679 **
4	2995	4771604	1	6070	3.8098	0.051046 .
5	2994	4770322	1	1283	0.8050	0.369682

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Degree-4 Polynomial



(Very) high order polynomials sometimes necessary to approximate functions and these are often still not sensible.

Step function

Polynomial imposes ‘global’ structure on the function of X .

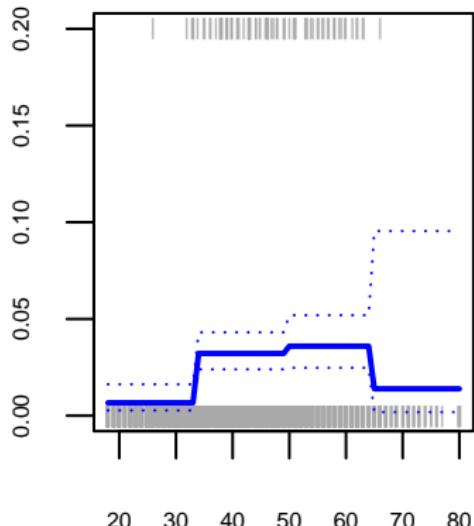
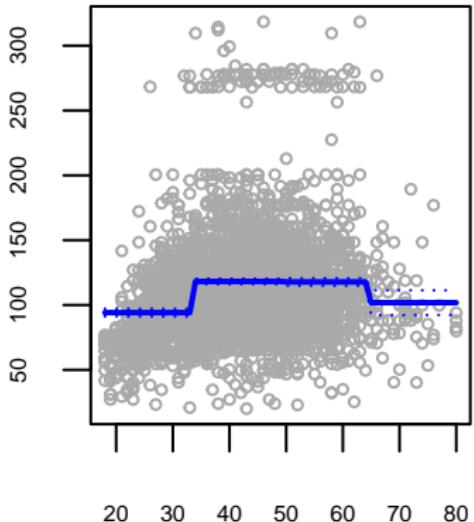
Step functions convert continuous variable to ‘ordered categorical’ variables.

Creates a piece-wise constant function.

$$y_i = \alpha + \beta_1 \mathbf{I}_1(x_i) + \beta_2 \mathbf{I}_2(x_i) + \beta_3 \mathbf{I}_3(x_i) + \epsilon_i$$

where \mathbf{I}_k is an indicator function returning 1 if x_i is within region k and 0 otherwise.

Step Function (K=4)



Piece-wise constant functions miss the action.
- fail to capture increasing wage with age.

Basis Functions

Polynomial and Step functions are special cases of basis function approach:

A family of functions or transformations that can be applied to a variable X .

e.g. $b_1(X)$, $b_2(X)$, $b_3(X)$, etc.

We can view predictions from a regression as a weighted sum of basis functions.

All inference tools for linear models are therefore available.

Basis functions must be chosen ahead of time (basis functions are known)

- other choices for constructing basis functions include wavelets & Fourier series.

Regression splines are a common choice for basis function.

Regression splines

Fit piecewise polynomials.

Divide data into $K + 1$ regions using K knots

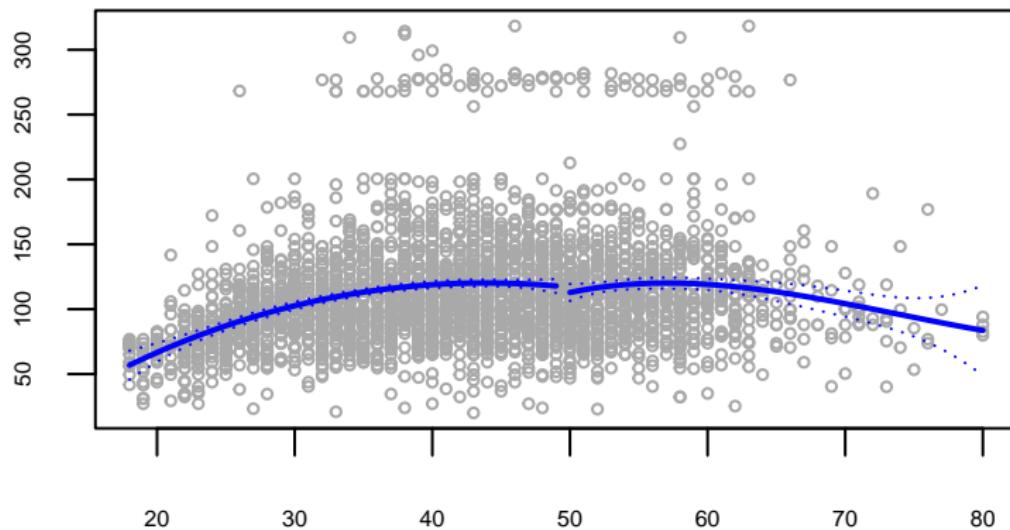
Separate polynomial fitted to the data in each region.

e.g. for a function with a single knot at $x = c$,

$$\hat{y}_i = \begin{cases} \text{polynomial 1 if } x_i < c \\ \text{polynomial 2 if } x_i \geq c \end{cases}$$

Immediate problem: functions are discontinuous and look silly!

piece-wise d=3 polynomial, knot at age=50



Constraints and splines

Add constraints:

First fitted curve must be continuous at $\text{age} = 50$.

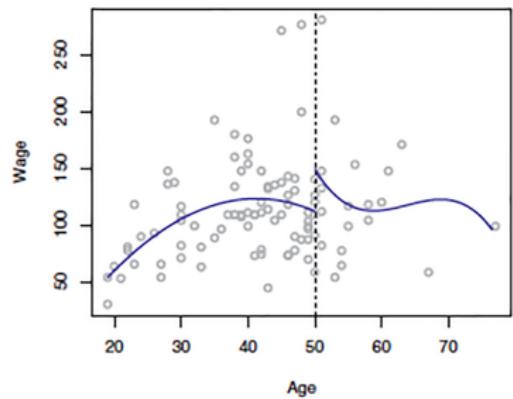
1st & 2nd derivatives are continuous at $\text{age} = 50$.

Cubic spline has $4 + K$ degrees of freedom (K is the number of knots)

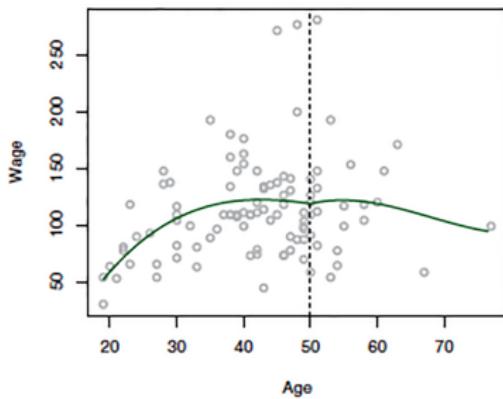
For a degree- d spline:

- piecewise degree- d polynomial with continuity in derivatives up to $d - 1$ at each knot.

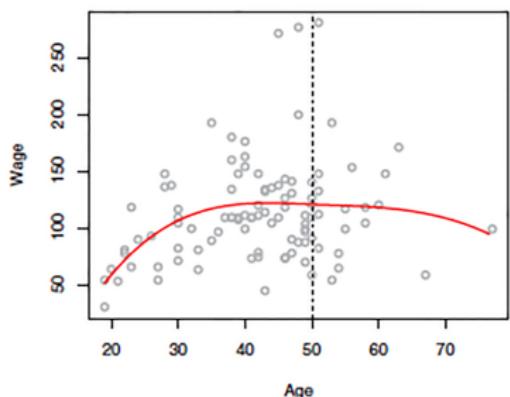
Piecewise Cubic



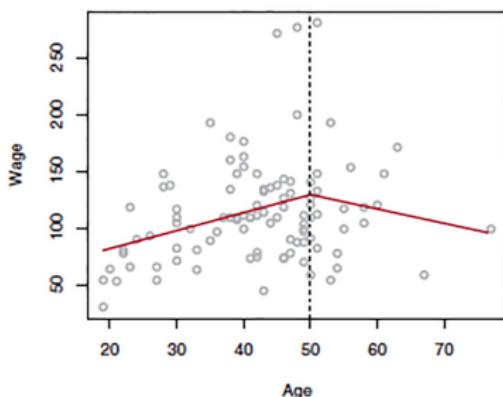
Continuous Piecewise Cubic



Cubic Spline



Linear Spline



Spline basis representation

How can we fit regression splines with constraints?

Use basis functions. e.g. for a cubic spline with K knots:

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

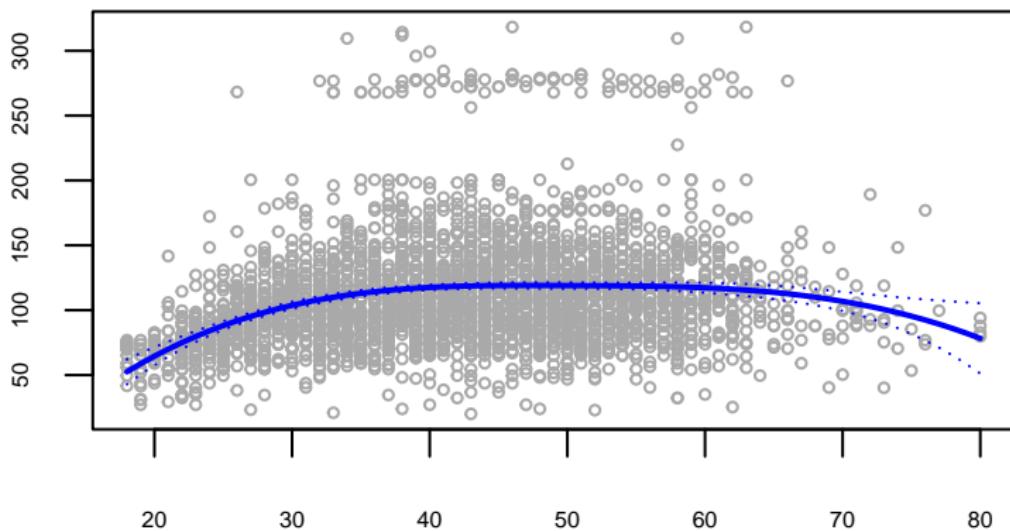
There are many ways to represent the basis functions for regression splines.

A common one is to add one *truncated power basis* per knot

$$h(x, \xi) = \begin{cases} (x - \xi)^3 & \text{if } x > \xi \\ 0 & \text{if otherwise,} \end{cases}$$

```
> Wage$trunc50 <- (Wage$age-50)^3  
> Wage$trunc50[Wage$age<50] <- 0  
> fit <- lm(wage ~ poly(age, 3) + trunc50, data=Wage)
```

cubic regression spline, 1 knot at age=50

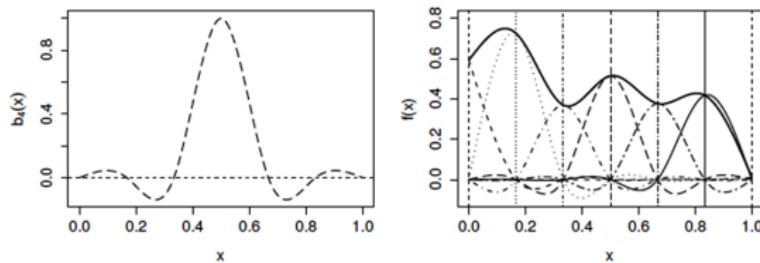


Spline basis representation

Truncated power basis can be unstable.

There are other ways of defining basis functions for cubic regression splines.

Some of these include parameterizations in terms of the values at the knots.



(Figure from Wood, 2006)

Others include:

- cyclic cubic regression splines
- B-splines (basis functions are strictly local)
- P-splines (for equally spaced knots)
- Thin plate splines (Duchon 1977), are an example of a smoothing spline.

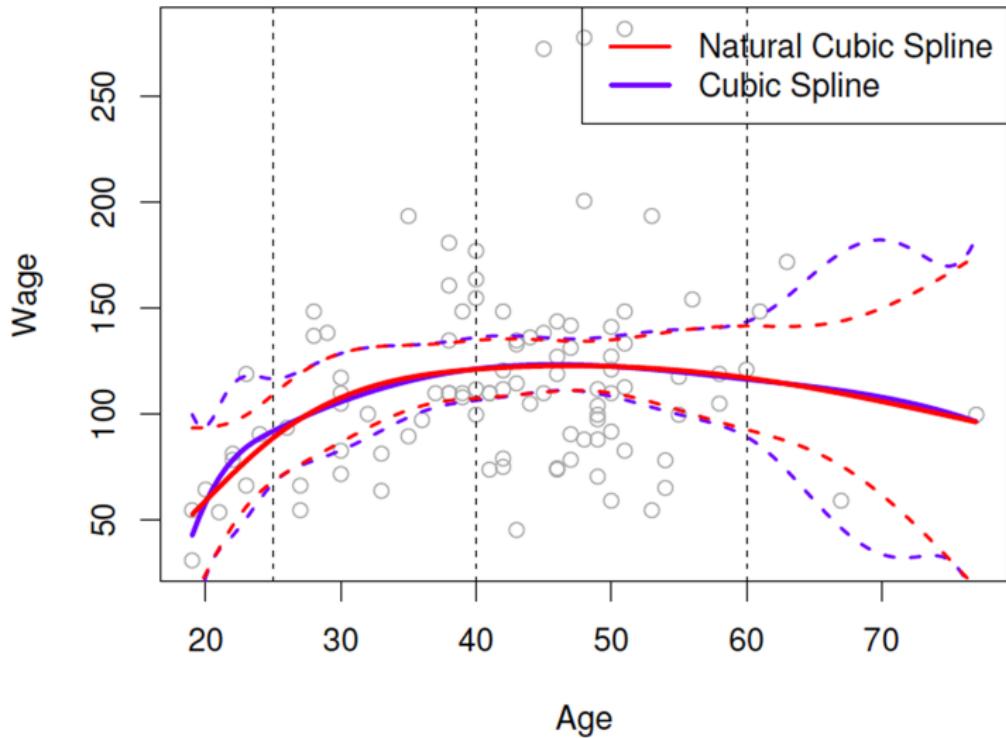
Natural splines

Splines can have high variance, particularly at the data extremes.

Natural splines have additional boundary constraints:

- the function is required to be linear outside the smallest/largest knots.

```
> fit <- lm(wage ~ ns(age, knots=c(25, 40, 60)),  
+             data=Wage)
```



Number & Location of knots

Might want to be flexible

- e.g. place more knots in region where suspect function most likely to be changing

More commonly, place knots in uniform fashion

- uniformly spaced over range of data
- uniform quantiles of data

In function calls:

- either request the degrees of freedom (e.g. 6df \implies 3 knots)
- or specify the knot locations.

```
> fit <- lm(wage ~ bs(age, knots=c(25, 40, 60)), data=Wage)
> fit2 <- lm(wage ~ bs(age, df=6), data=Wage)
> attr(bs(age,df=6),"knots")
[1] 33.75 42.00 51.00
```

Note that if df= is chosen then knots are spaced at uniform quantiles of the data.

Number & Location of knots

Run for different numbers of knots, look at which produces the best curve.

Cross-validation more objective.

1. remove $k\%$ of the data. (k -fold cv)
2. fit spline to remaining data
3. Predict the left-out data, compute RSS (or other measure of fit)
4. Repeat 1-3 until all data used, compute total RSS
5. Repeat 1-4 using different # of knots

Splines v polynomials:

Polynomials require high degree to produce flexible fits.

Splines, increase # knots but keep degree fixed.

- generally produce more stable estimates.
- also allow knots to be placed where function appears to be changing most rapidly.

Smoothing splines

Related to regression splines but instead have n knots, 1 for each data point.

In general, we want $\sum_i (y_i - g(x_i))^2$ to be small.

Could make $g()$ **so** flexible that it interpolates all the y_i .

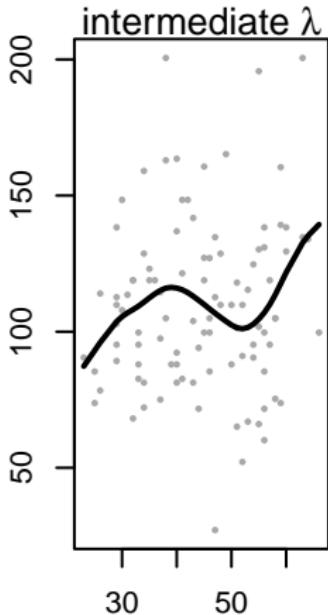
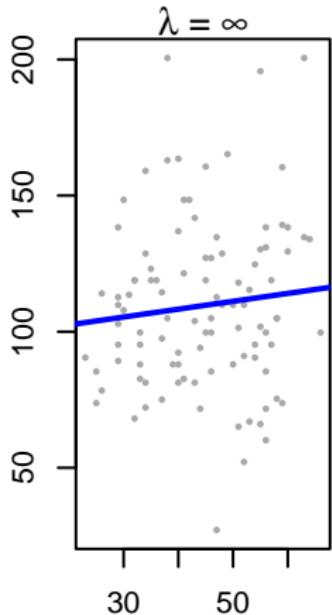
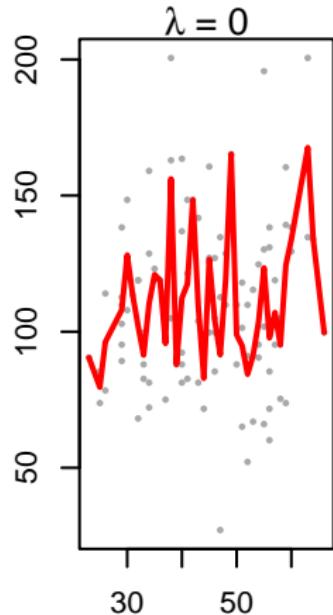
Find function g that minimizes

$$\sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int_0^\infty g''(t)^2 dt$$

'loss' plus 'penalty'

The tuning parameter λ controls the bias-variance tradeoff of smoothing spline.

Lambda controls the smoothness



(Kenny G has a high λ)

Choosing λ

If $g()$ is a piece-wise cubic polynomial with knots at x_1, x_2, \dots, x_n , with continuous 1st & 2nd derivatives at each knot, then it is a 'shrunken' version of a natural cubic spline.

(λ controls the shrinkage)

$$\text{As } \begin{matrix} \lambda & 0 & \rightarrow & \infty \\ df & n & \rightarrow & 2 \end{matrix}$$

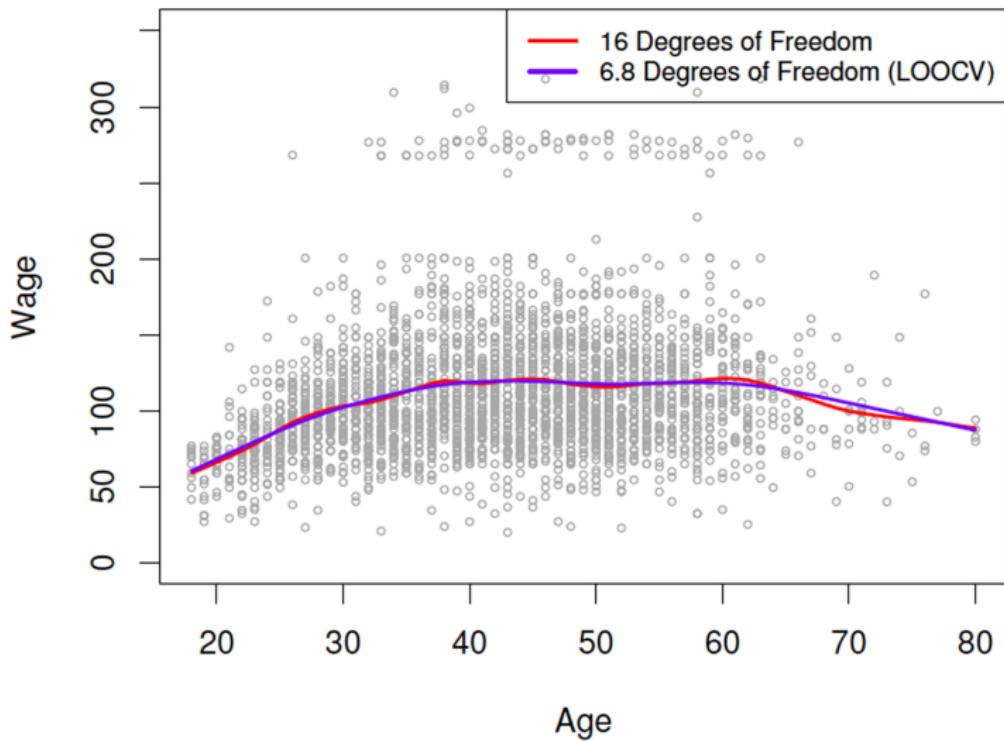
Effective degrees of freedom

$$df_\lambda = \sum_{i=1}^n S_{\lambda i} i \text{ (sum of diagonal)}$$

Use cross-validation to choose the λ that minimizes the cumulative cross-validated RSS.

$$\text{RSS}_{cv}(\lambda) = \sum_{i=1}^n (y_i - \hat{g}_\lambda^{(-i)}(x_i))^2 = \sum_{i=1}^n \left[\frac{y_i - \hat{g}_\lambda(x_i)}{1 - \{\mathbf{S}_\lambda\}_{ii}} \right]^2$$

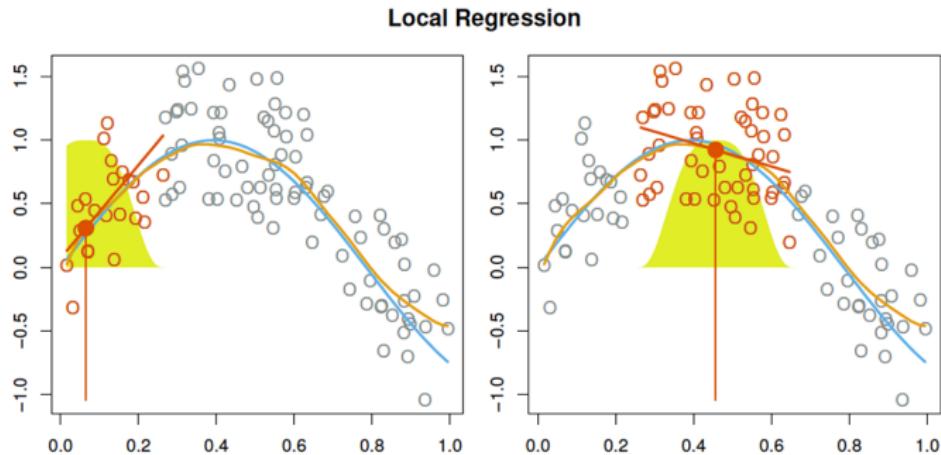
Smoothing Spline



Local regression

Fit at a given point computed using only neighboring data.

A new weighted least squares regression at each point.



LOESS smooth probably most common.
(tri-cube weight function)

Local regression

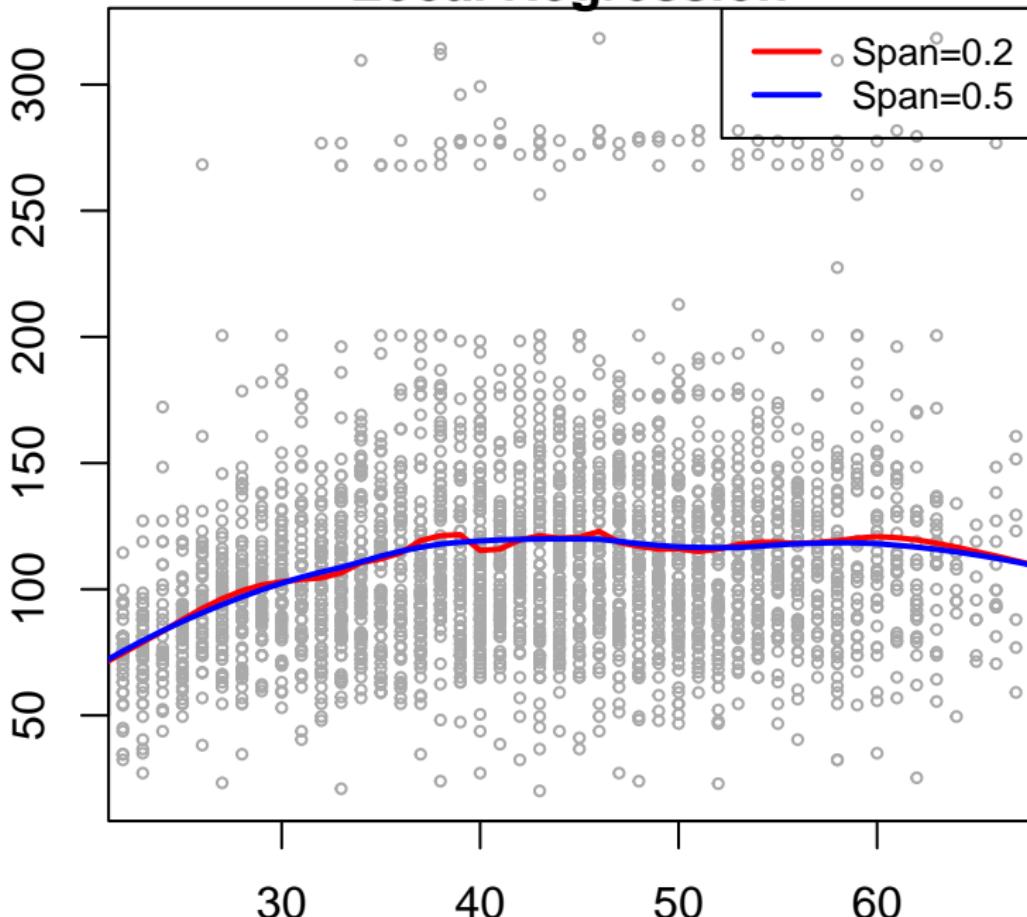
Basis functions include the weights assigned to neighboring data points.

$$\hat{Y}_{40} = \begin{pmatrix} 0 & \cdots & 0 & 0.1 & 0.2 & 0.4 & 0.2 & 0.1 & 0 & \cdots & 0 \end{pmatrix} \begin{pmatrix} Y_1 \\ \vdots \\ Y_{37} \\ Y_{38} \\ Y_{39} \\ Y_{40} \\ Y_{41} \\ Y_{42} \\ Y_{43} \\ \vdots \\ Y_{100} \end{pmatrix} = s \times Y$$

e.g.

Most important choice is the *span*, which controls the size of the neighborhood and therefore the wiggleness.

Local Regression



Generalized Additive Modeling

(asynchronous on Thu)

Uses the methods today as building blocks for fitting an additive model.

Additional Recommended Reading

Wood, S., 2006. Generalized additive models: an introduction with R. CRC press.

(today's material Chapters 3 & 4.1)

Hastie, T.J. and Tibshirani, R.J., 1990. Generalized additive models (Vol. 43). CRC Press.

Wahba, G. (1990). Spline Models for Observational Data. Philadelphia: Society for Industrial and Applied Mathematics.