Module 7: Moving Beyond Linearity

TMA4268 Statistical learning

Thiago G. Martins

February, 2021

Multiple linear regression

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots \beta_k x_{ik} + \varepsilon_i,$$

or equivalently

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon =$$

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Estimation

The OLS estimator for β is

$$\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}.$$

We will now change $\mathbf{X}$ as we like, but keep $\hat{\beta}$.

Non-Linear Models

Let us focus on **one** explanatory variable X for now. We will generalize later.

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots \beta_k b_k(x_i) + \varepsilon_i,$$

where $b_j(x_i)$ are **basis functions**.

Example with $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i$

We have

$$b_1(X) = X,$$

$$b_2(X) = X^2,$$

$$\mathbf{x} = (6, 3, 6, 8)^\mathsf{T},$$

$$\mathbf{y} = (3, -2, 5, 10)^\mathsf{T}.$$

This results in

$$\mathbf{X} = \begin{pmatrix} 1 & b_1(x_1) & b_2(x_1) \\ 1 & b_1(x_2) & b_2(x_2) \\ 1 & b_1(x_3) & b_2(x_3) \\ 1 & b_1(x_4) & b_2(x_4) \end{pmatrix} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \\ 1 & x_4 & x_4^2 \end{pmatrix} = \begin{pmatrix} 1 & 6 & 36 \\ 1 & 3 & 9 \\ 1 & 6 & 36 \\ 1 & 8 & 64 \end{pmatrix}$$

and

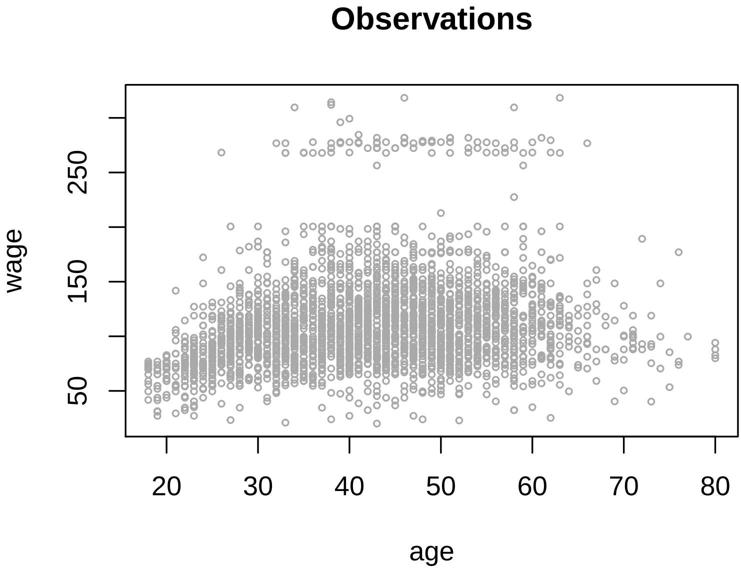
$$\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y} = (-4.4, 0.2, 0.2)^\mathsf{T}.$$

General Design Matrix

$$\mathbf{X} = \begin{pmatrix} 1 & b_1(x_1) & b_2(x_1) & \dots & b_k(x_1) \\ 1 & b_1(x_2) & b_2(x_2) & \dots & b_k(x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & b_1(x_n) & b_2(x_n) & \dots & b_k(x_n) \end{pmatrix}.$$

- Rows are observations
- Columns are basis functions
- Same setup as for multiple linear regression

The Aim



Use `lm(wage ~ X)` and choose **X** according to method.

Polynomial Regression

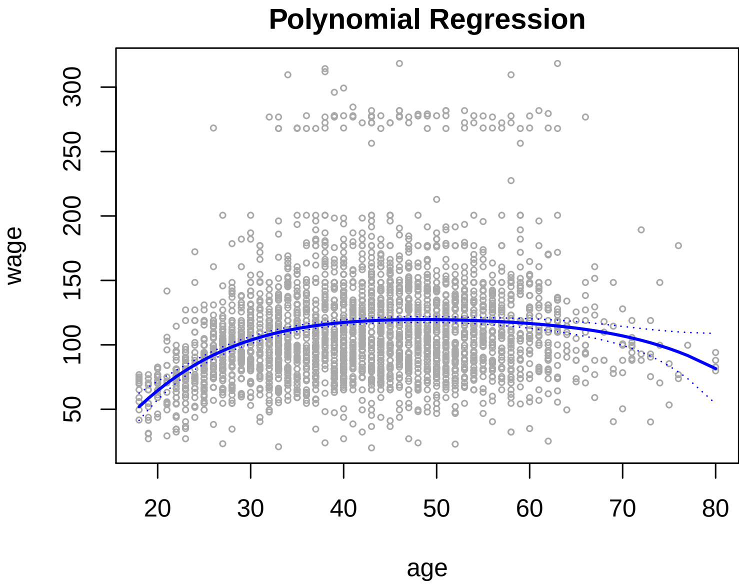
The polynomial regression includes powers of X in the regression.

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ldots \beta_k x_i^d + \varepsilon_i,$$

- In practice $d \leq 4$
- The basis is $b_j(x_i) = x_i^j$ for $j = 1, 2 \ldots, d$

$$\mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^d \\ 1 & x_2 & x_2^2 & \ldots & x_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^d \end{pmatrix}.$$

```r
fit = lm(wage ~ poly(age,4))
Plot(fit, main = "Polynomial Regression")
```



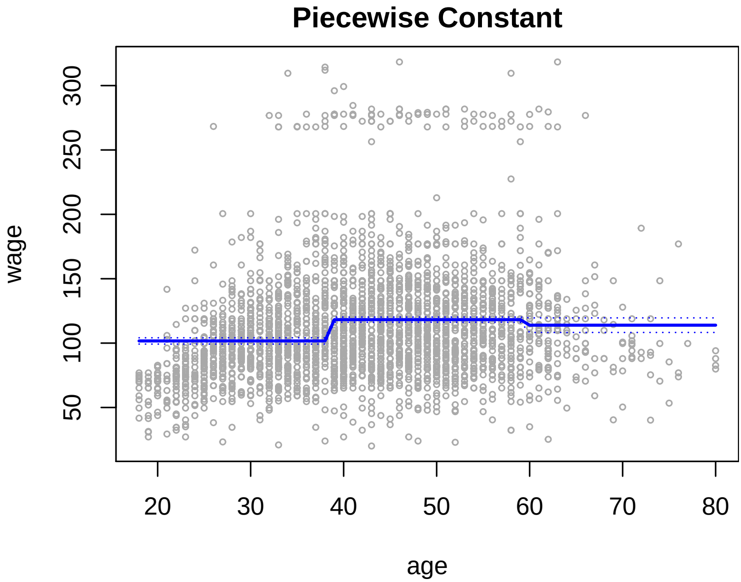
Step Functions

- Divide age into bins
- Model `wage` as a constant in each bin
- The basis functions indicate which bin x_i belongs to
- Cutpoints $c_1, c_2, \ldots, c_K$

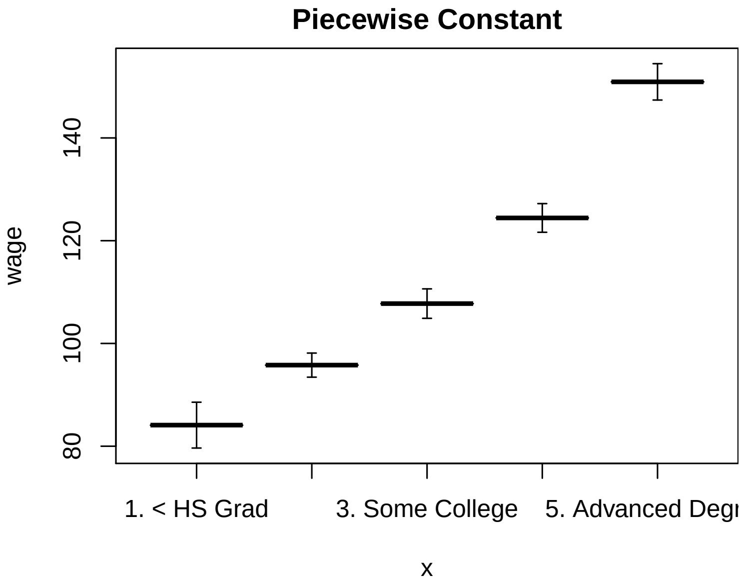
$$b_j(x_i) = I(c_j \leq x_i < c_{j+1})$$

$$\mathbf{X} = \begin{pmatrix} 1 & I(x_1 < c_1) & I(c_1 \leq x_1 < c_2) & \ldots & I(c_K \leq x_1) \\ 1 & I(x_2 < c_1) & I(c_1 \leq x_2 < c_2) & \ldots & I(c_K \leq x_2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & I(x_n < c_1) & I(c_1 \leq x_n < c_2) & \ldots & I(c_K \leq x_n) \end{pmatrix}.$$

```r
fit = lm(wage ~ cut(age,3))
Plot(fit, main = "Piecewise Constant")
```



```r
fit = lm(wage ~ education)
Plot(fit, main = "Piecewise Constant")
```

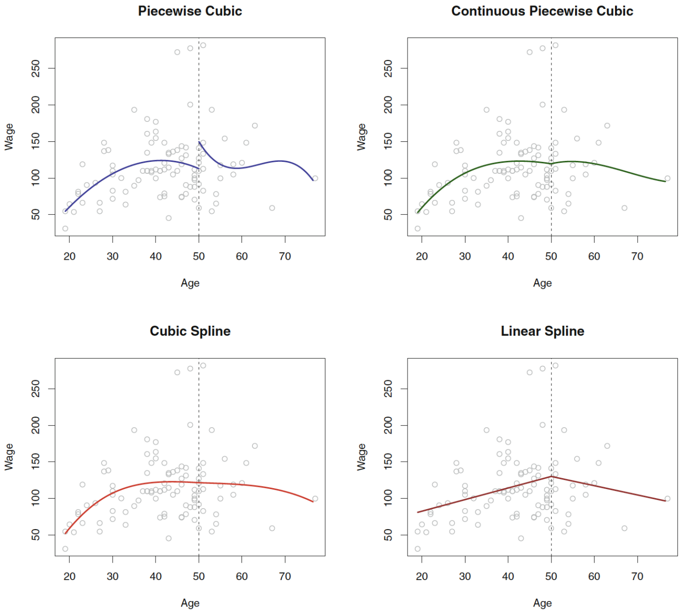


Regression Splines

A degree-d spline is a piecewise degree-d polynomial, with continuity in derivatives up to degree $d-1$ at each knot.

- Combination of polynomials and step functions
- **Knots** $c_1, c_2, \ldots, c_K$
- Continous derivatives up to order $d-1$ at each knot.

Regression Splines



Regression Splines

An order-d spline with K knots has the basis

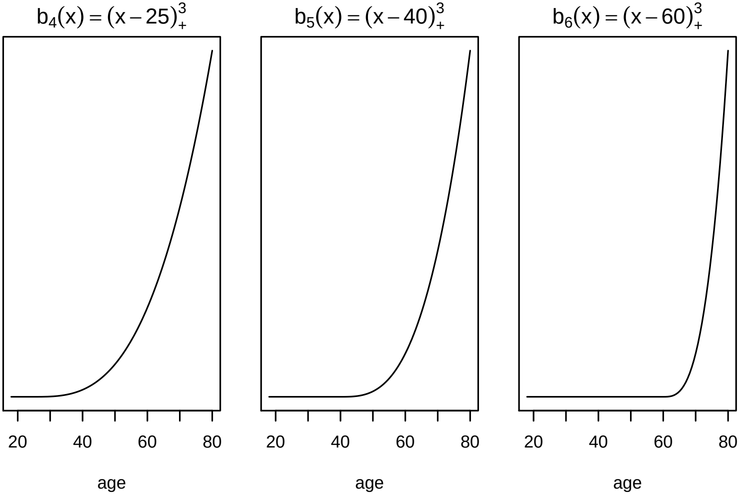
$$\begin{aligned} b_j(x_i) &= x_i^j &&, \ j = 1, \ldots, d \\ b_{d+k}(x_i) &= (x_i - c_k)_+^d &&, \ k = 1, \ldots, K, \end{aligned}$$

where

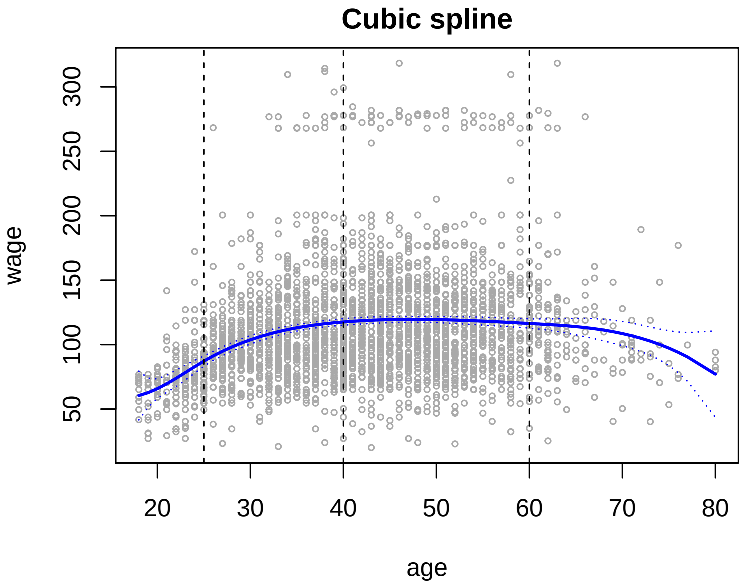
$$(x - c_j)_+^d = \begin{cases} (x - c_j)^d & , x > c_j \\ 0 & , \text{otherwise.} \end{cases}$$

Cubic Splines

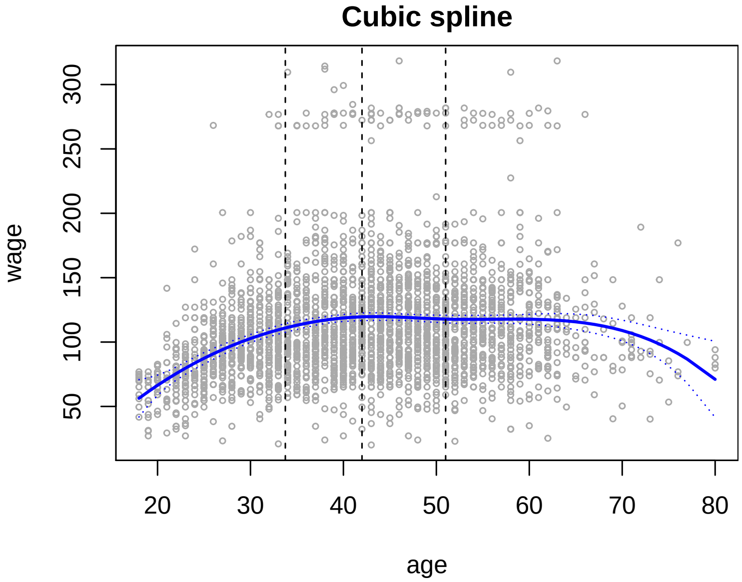
- A spline with $d = 3$ is cubic
- The basis is $X, X^2, X^3, (X - c_1)^3_+, (X - c_2)^3_+, \ldots, (X - c_K)^3_+$



```r
fit = lm(wage ~ bs(age, knots = c(25,40,60)))
Plot(fit, main = "Cubic spline")
```



```r
fit = lm(wage ~ bs(age, df = 6))
Plot(fit, main = "Cubic spline")
```



Natural Cubic Splines

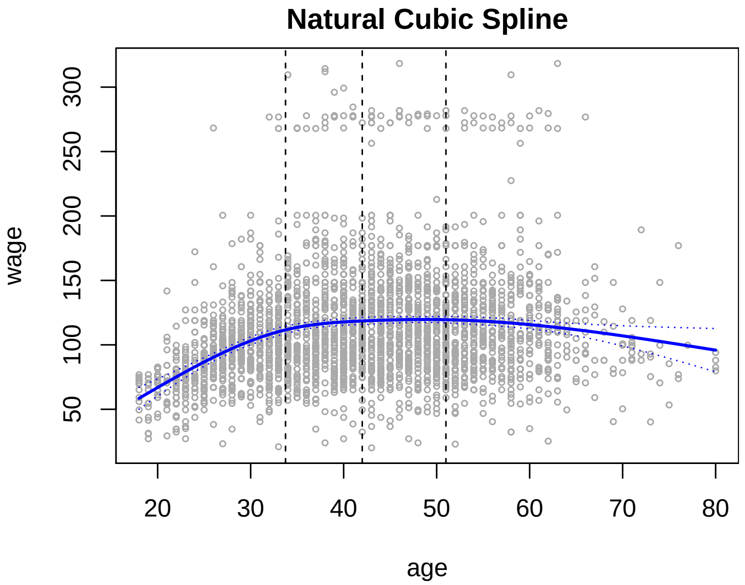
- Cubic spline that is linear at the ends
- The idea is to reduce variance
- Straight line outside $c_0 = 18$ and $c_{K+1} = 80$
- We call these points **boundary knots**

The basis is

$$b_1(x_i) = x_i, \quad b_{k+2}(x_i) = d_k(x_i) - d_K(x_i),\ k = 0, \ldots, K-1,$$

$$d_k(x_i) = \frac{(x_i - c_k)_+^3 - (x_i - c_{K+1})_+^3}{c_{K+1} - c_k}.$$

```r
fit = lm(wage ~ ns(age, df = 4))
Plot(fit, main = "Natural Cubic Spline")
```



Recap

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon.$$

- Non-linear methods, but linear regression.
- Each method defined by a basis, $\mathbf{X}_{ij} = b_j(x_i)$.
- And simply $\hat{\beta} = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$
- We will now move from $\mathbf{X}\beta$ to $f(X)$

$$\mathbf{y} = \mathbf{f}(X) + \varepsilon.$$

Smoothing Splines

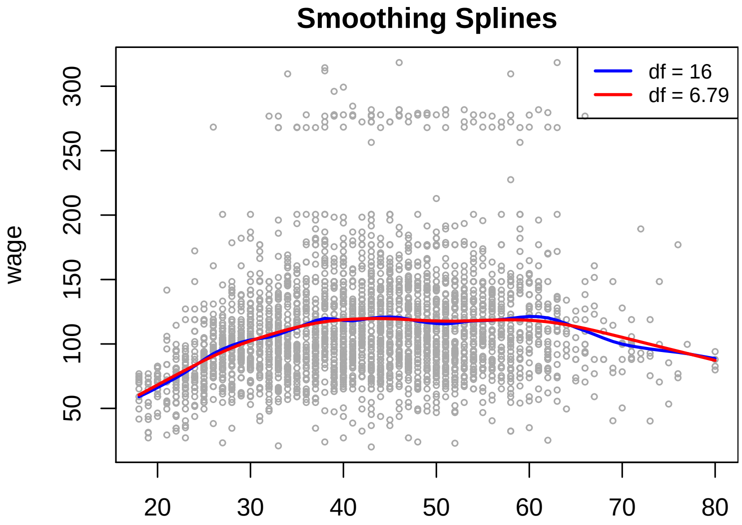
- Different idea than regression splines
- Minimize the prediction error
- Bias-variance approach

A smoothing spline is the function g that minimizes

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 \mathrm{d}t.$$

- What happens as $\lambda \to \infty$?
- What happens as $\lambda \to 0$?

```r
fit = smooth.spline(age, wage, df = 16)
Plot(fit, main = "Smoothing Splines")
fit = smooth.spline(age, wage, cv = T)
Plot(fit, legend = 16)
```



The Smoother Matrix

The fitted values are

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

The effective degrees of freedom is

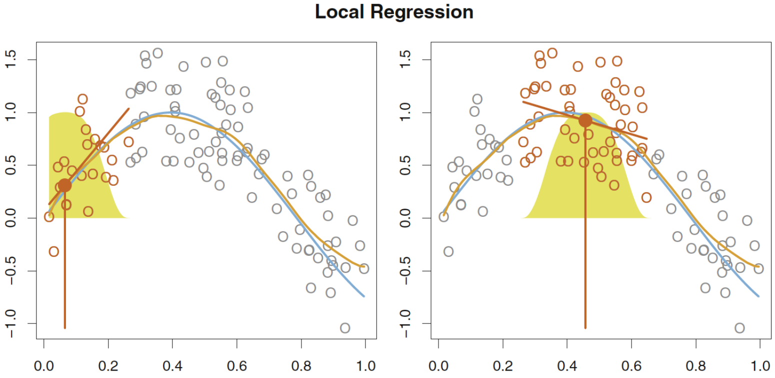
$$df_\lambda = \mathrm{tr}(\mathbf{S}).$$

The leave-one-out cross-validation error is

$$\mathrm{RSS}_{cv}(\lambda) = \sum_{i=1}^{n} \left(\frac{y_i - \hat{y}_i}{1 - \mathbf{S}_{ii}} \right)^2 .$$

Local Regression

- Smoothed k-nearest neighbor algorithm
- Run for each x_0
- Draw a line $\beta_0 + \beta_1 x$ through neighborhood
- Close observations are weighted more heavily
- The fitted value is $\hat{\beta}_0 + \hat{\beta}_1 x_0$



Local Regression

Finding the $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize

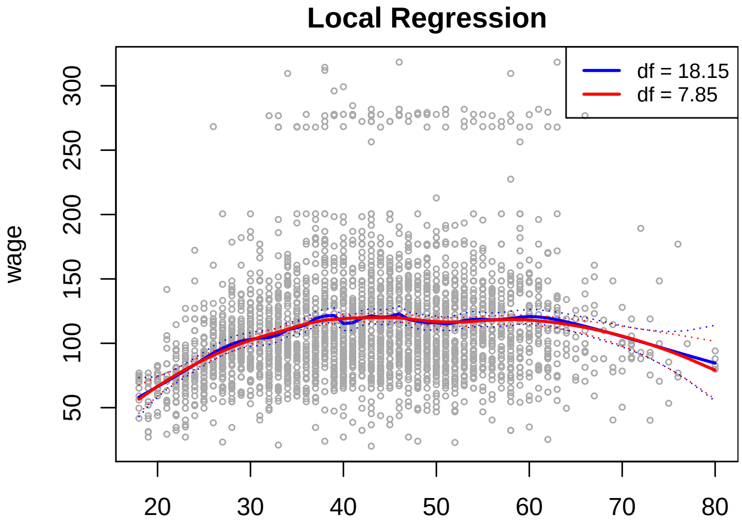
$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2,$$

where

$$K_{i0} = \left(1 - \left|\frac{x_0 - x_i}{x_0 - x_\kappa}\right|^3\right)_+^3.$$

Local Regression

```r
fit = loess(wage ~ age, span = .2)
Plot(fit, main = "Local Regression")
Plot(loess(wage ~ age, span=.5),legend=fit$trace.hat)
```



Additive Models

Combines the models we have discussed so far. For example

$$
\begin{aligned}
y_i &= f_1(x_{i1}) + f_2(x_{i2}) + \varepsilon_i \\
&= f(x_i) + \varepsilon_i.
\end{aligned}
$$

If each component is on the form $\mathbf{X}\beta$, so is f.

Component 1

- Cubic spline with $X_1 = \text{age}$
- Knots at 40 and 60

The design matrix when excluding the intercept is

$$\mathbf{X}_1 = \begin{pmatrix} x_{11} & x_{11}^2 & x_{11}^3 & (x_{11} - 40)_+^3 & (x_{11} - 60)_+^3 \\ x_{21} & x_{21}^2 & x_{21}^3 & (x_{21} - 40)_+^3 & (x_{21} - 60)_+^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{n1} & x_{n1}^2 & x_{n1}^3 & (x_{n1} - 40)_+^3 & (x_{n1} - 60)_+^3 \end{pmatrix}.$$

Component 2

- Natural spline with $X_2 = \texttt{year}$
- Knot at $c_1 = 2006$
- Boundary knots at $c_0 = 2003$ and $c_2 = 2009$

The design matrix when excluding the intercept is

$$\mathbf{X}_2 = \begin{pmatrix} x_{12} & \left[\frac{1}{6}(x_{12} - 2003)^3 - \frac{1}{3}(x_{12} - 2006)^3_+\right] \\ x_{22} & \left[\frac{1}{6}(x_{22} - 2003)^3 - \frac{1}{3}(x_{22} - 2006)^3_+\right] \\ \vdots & \vdots \\ x_{n2} & \left[\frac{1}{6}(x_{n2} - 2003)^3 - \frac{1}{3}(x_{n2} - 2006)^3_+\right] \end{pmatrix}.$$

Component 3

- Factor $X_3 =$ `education`
- Levels `< HS Grad`, `HS Grad (HSG)` , `Some College (SC)` , `College Grad (CG)` and `Advanced Degree (AD)`
- Dummy variable coding

The design matrix when excluding the intercept is

$$\mathbf{X}_3 = \begin{pmatrix} I(x_{13} = \mathrm{HSG}) & I(x_{13} = \mathrm{SC}) & I(x_{13} = \mathrm{CG}) & I(x_{13} = \mathrm{AD}) \\ I(x_{23} = \mathrm{HSG}) & I(x_{23} = \mathrm{SC}) & I(x_{23} = \mathrm{CG}) & I(x_{23} = \mathrm{AD}) \\ \vdots & \vdots & \vdots & \vdots \\ I(x_{n3} = \mathrm{HSG}) & I(x_{n3} = \mathrm{SC}) & I(x_{n3} = \mathrm{CG}) & I(x_{n3} = \mathrm{AD}) \end{pmatrix}.$$

Additive Model

Combine the components to

$$\mathbf{y}_i = f_1(x_{i1}) + f_2(x_{i2}) + f_3(x_{i3}) + \varepsilon_i.$$

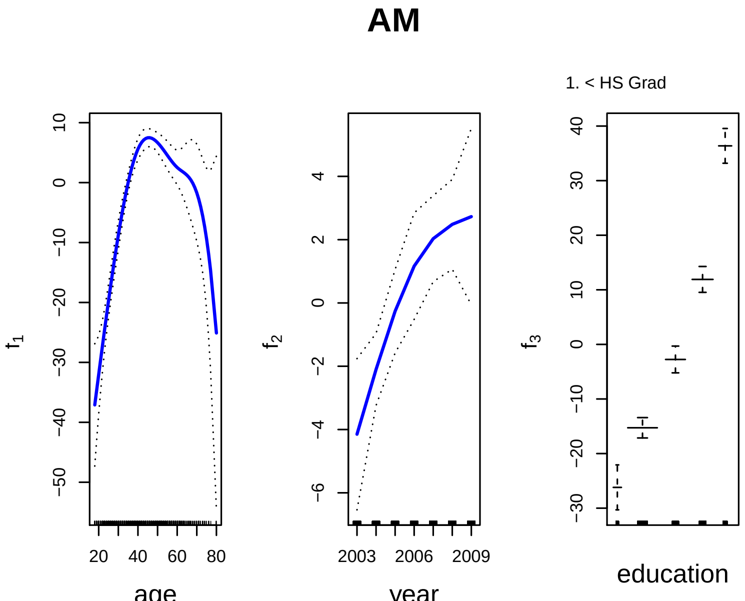
Since each component is linear, we can write

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon,$$

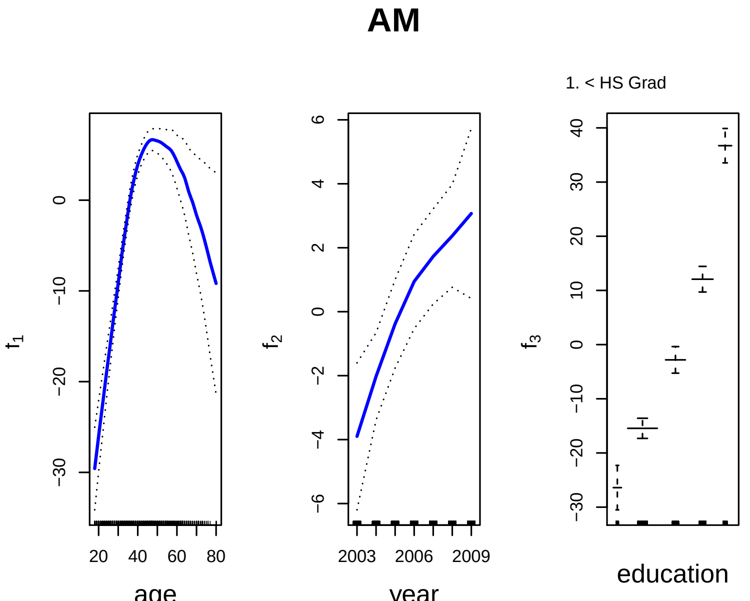
where

$$\mathbf{X} = \begin{pmatrix} \mathbf{1} & \mathbf{X}_1 & \mathbf{X}_2 & \mathbf{X}_3 \end{pmatrix}.$$

```r
fit = gam(wage ~ bs(age, knots = c(40, 60)) +
            ns(year,knots=2006) + education)
Plot(fit)
```



```r
fit = gam(wage ~ lo(age, span = 0.6) +
              s(year, df = 2) + education)
Plot(fit)
```



Qualitative Responses

- Logistic regression
- $Y = 0$ or $Y = 1$
- $p(X) = \Pr(Y = 1|X)$

The generalized logistic regression model is

$$\log\left(\frac{p(X)}{1 - p(X)}\right) = f(X).$$

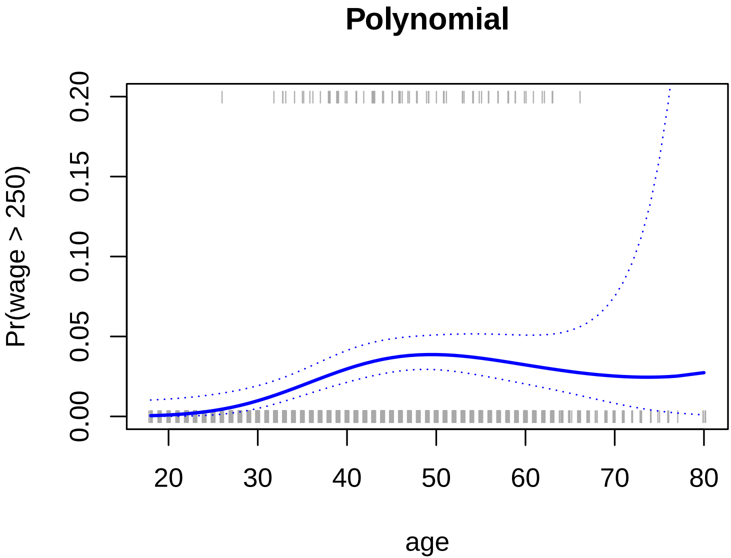
Choose f from the methods we have learned.

Polynomial Logistic Regression

With degree 4 we have

$$\log\left(\frac{p(X_1)}{1-p(X_1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \beta_4 X_1^4.$$

```r
fit = glm(I(wage>250) ~ poly(age,3),
          family = "binomial")
Plot(fit, main = "Polynomial")
```

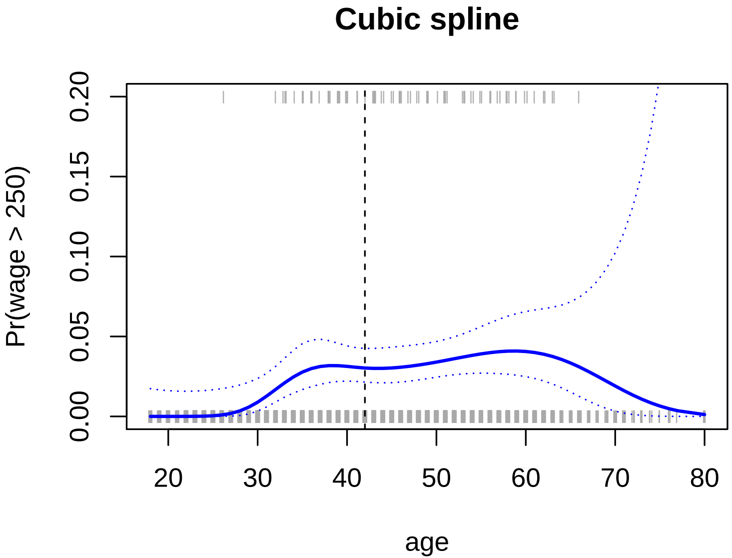


Cubic Spline Logistic Regression

- A cubic spline in age
- Knot at 42

$$\log\left(\frac{p(X_1)}{1-p(X_1)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3 + \beta_4 (X_1 - 42)_+^3.$$

```r
fit = glm(I(wage>250) ~ bs(age, df = 4),
          family = "binomial")
Plot(fit, main = "Cubic spline")
```

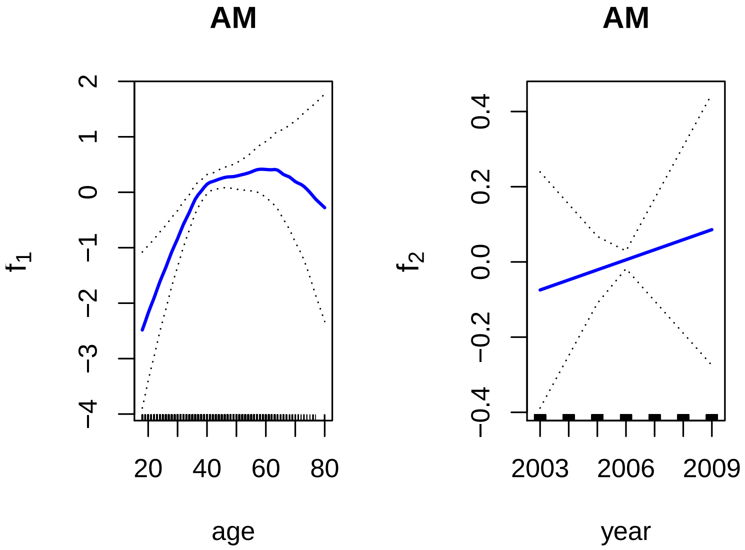


GAM

- f_1 is a local regression in `age`
- f_2 is a simple linear regression in `year`

$$\log\left(\frac{p(X_1, X_2)}{1 - p(X_1, X_2)}\right) = \beta_0 + f_1(X_1) + f_2(X_2).$$

```r
par(mfrow=c(1,2))
Plot(gam(I(wage>250) ~ lo(age, span = 0.6) + year,
         family = "binomial"), multi = T)
```



References

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics New York.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 112. Springer.

Reinsch, Christian H. 1967. "Smoothing by Spline Functions." *Numerische Mathematik* 10 (3): 177–83.

Rodrıguez, German. 2001. "Smoothing and Non-Parametric Regression."

Tibshirani, Ryan, and L Wasserman. 2013. "Nonparametric Regression." Statistical Machine Learning, Spring.