Chapter 7: Moving Beyond Linearity

- Polynomial regression. Add powers of a predictor, e.g. x^2, x^3, to a simple linear model.
- Step functions. Range is divided into K distinct regions.
- Regression splines. A mixture of the two above. Range of X divided into K regions and then fit with polynomials and joined smoothly at knots.
- Smoothing splines. Similar to regression splines but involve using a smoothness penalty.
- Local regression. Similar to splines except regions may overlap.
- Generalized additive models. Extend methods above using multiple predictors.

Polynomial Regression

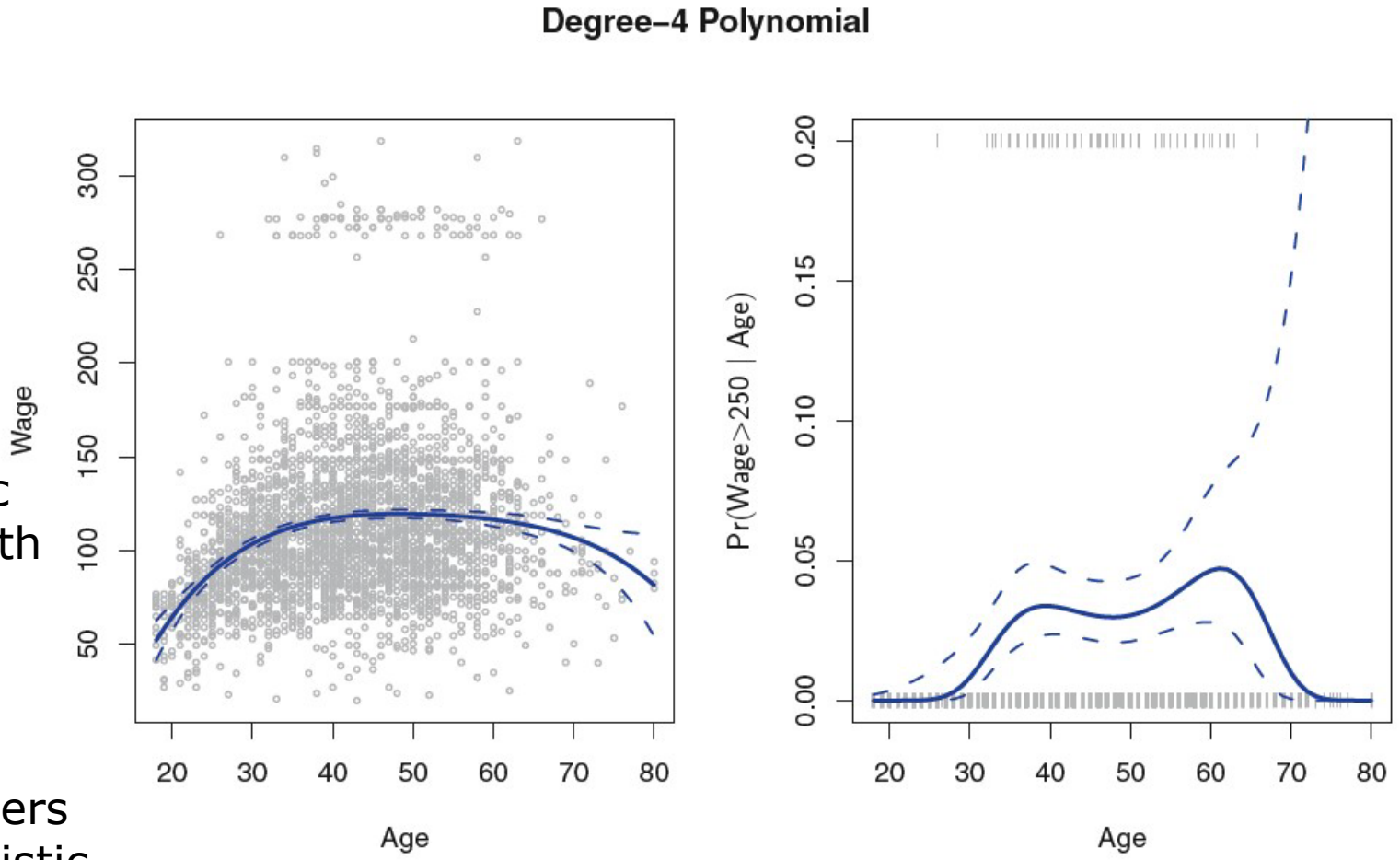
- The traditional of changing a linear model, $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, is to add powers of x_i, e.g. $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \epsilon_i$.
- This last equation is a polynomial of degree *d*.
- After using least squares to estimate the regression coefficients we can make predictions and put a $\pm 2s.e$ interval around those predictions.
- For a particular value of *x*, x_0, let $l_0^T = \left(1, x_0, x_0^2, \cdots, x_o^d\right)$, then $\hat{f}(x_0) = l_0^T \hat{\beta}, and\ Var(\hat{f}(x_0)) = l_0^T \hat{S} l_0$, where $\hat{S}$ is the variance/covariance matrix of $\hat{\beta}$.

Polynomial Regression

The wage data on the left shows a fourth order polynomial fit to wages as a function of age. The dashed lines are 95% confidence intervals.

The right figure shows the logistic regression function fit with a fourth order polynomial to predict the probability of earning a wage > $250,000.

Since there are only 79 high earners the confidence interval on the logistic regression results are wide.



Step Functions

- Create K cutpoints, c_1, c_2, ..,c_K and then use these to generate K+1 variables,

$$C_0(X) = I(X < c_1),$$
$$C_j(X) = I\left(c_j \leq X < c_{j+1}\right)$$
$$C_K(X) = I(c_K \leq X)$$

- The indicator variable, $I()$, is 1 if the condition is true and 0 otherwise.
- The regression model is then

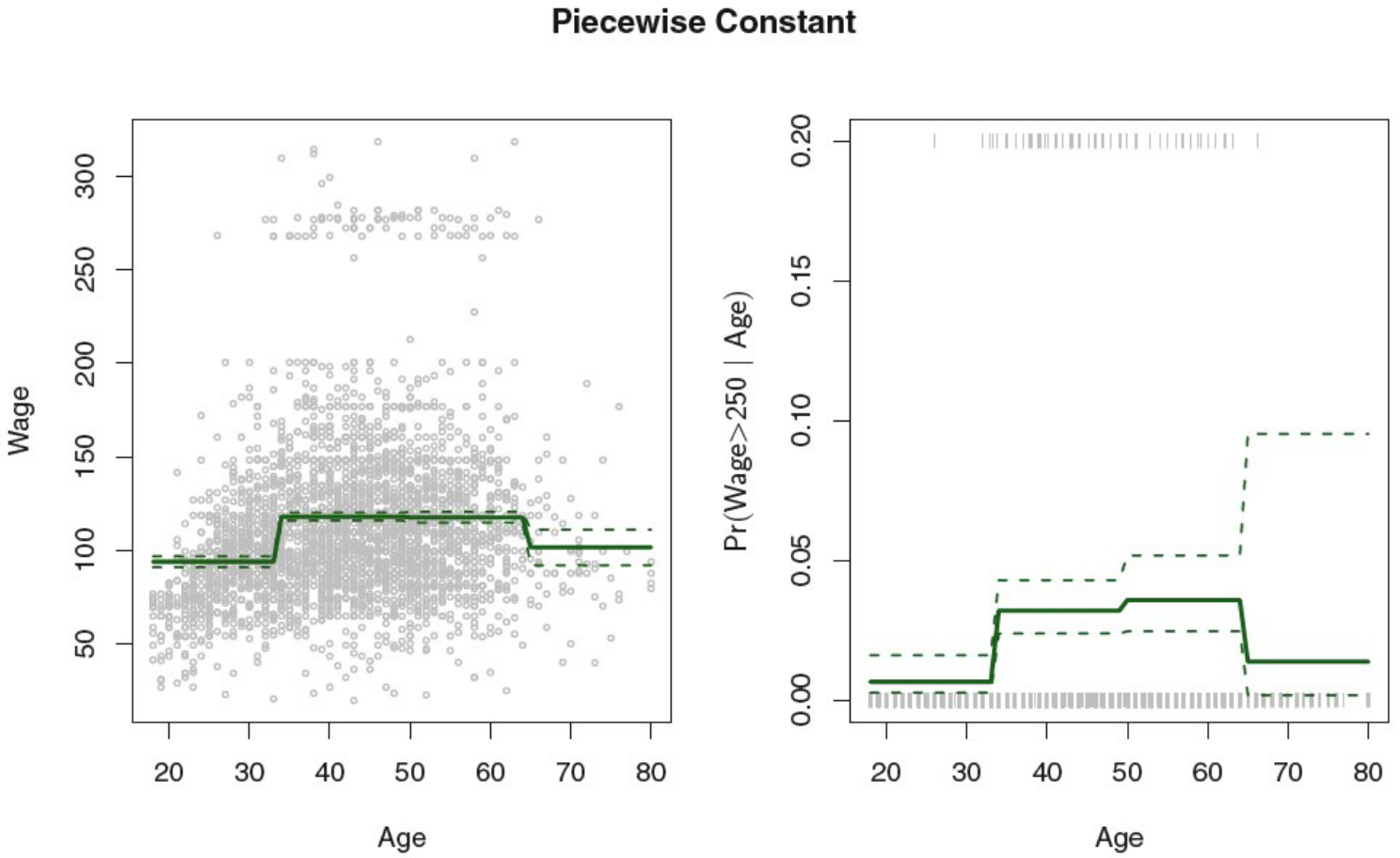
$$y_i = \beta_0 + \beta_1 C_1(x_i) + \cdots + \beta_K C_K(x_i) + \epsilon_i$$

- C_0 is replaced by β_0 in this equation.

Step Functions

These step functions have limited utility for capturing the early increase in wages.

Perhaps it would do better if the knot position could be estimated.



The same wage data with fitted step functions and 95% confidence intervals

Basis Functions

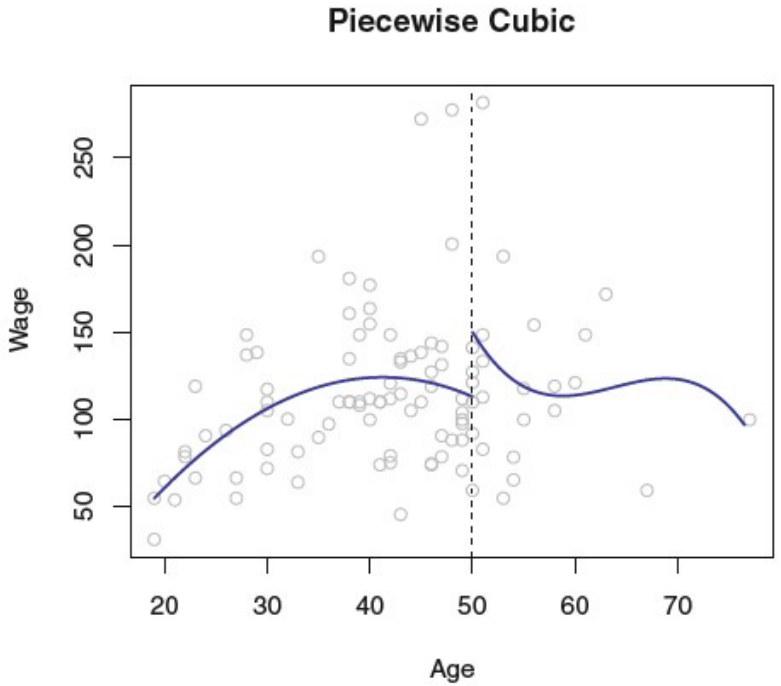
- Polynomial and step function regression are special cases of basis functions. That is functions that are pre-selected to use in linear models, e.g. $y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_K b_K(x_i) + \epsilon_i$.
- In the case of polynomial regression, $b_j(x_i) = x_i^j$, for piecewise constant functions, $b_j(x_i) = I(c_j \leq x_i < c_{j+1})$
- Next, we examine another basis function, regression splines.

Regression Splines

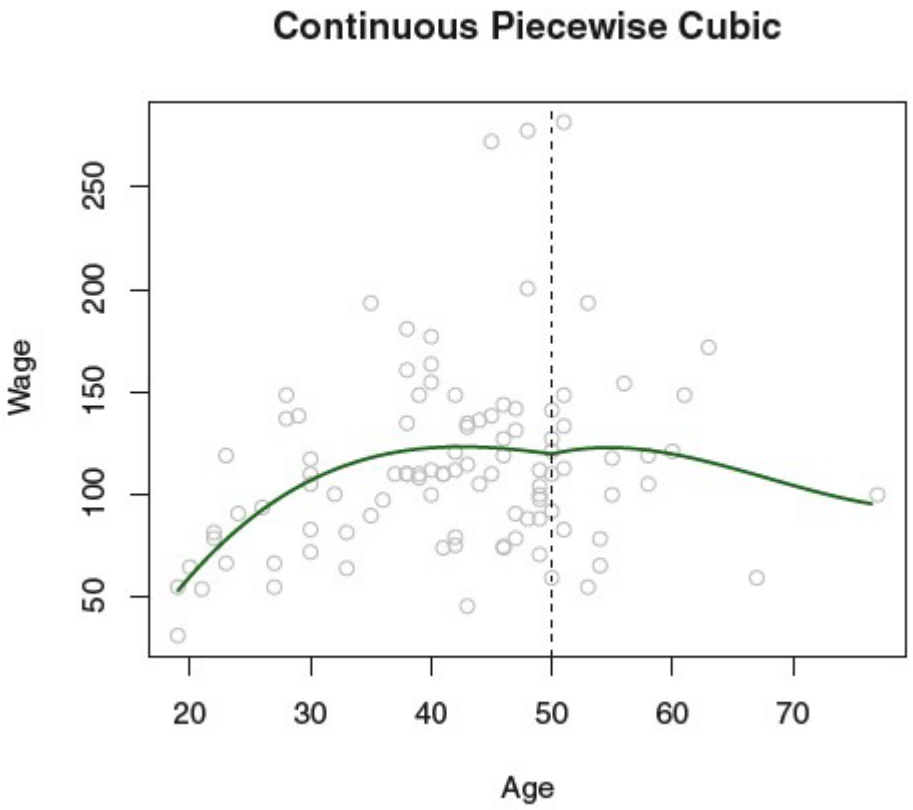
- For instance, we could also do a piecewise, polynomial regression, where we have cutpoints or knots. In each interval we fit a, say, cubic polynomial, $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \epsilon_i$.
- If there was only a single knot at c, then the model would be,

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

- This function fit to the wage data shows a discontinuity at the knot. Even with the eight degrees of freedom used to make these estimates.



Regression Splines

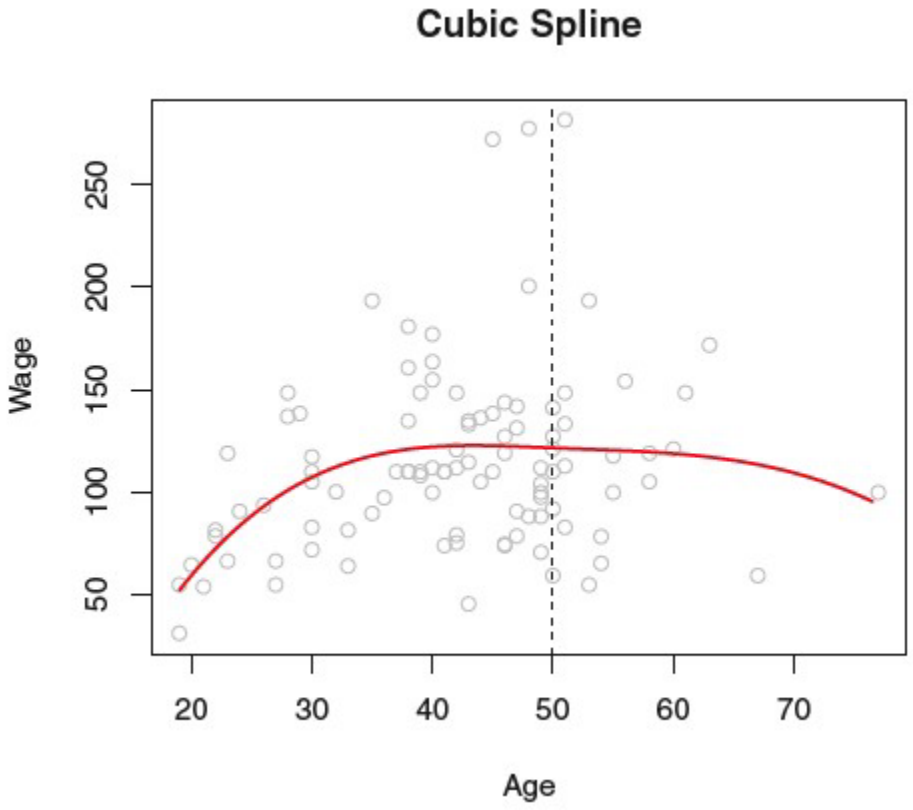


To make the splines continuous at the knot this figure adds the constraint,

$\beta_{01} + \beta_{11}c + \beta_{21}c^2 + \beta_{31}c^3 = \beta_{02} + \beta_{12}c + \beta_{22}c^2 + \beta_{33}c^3$

at $x=c$, which is 50 in the figure above.

A cubic spline will have 4+K degrees of freedom.



To make the knot smooth we next set the first and second derivative at the know to be equal, e.g.

$\beta_{11} + 2\beta_{21}c + 3\beta_{31}c^2 = \beta_{12} + 2\beta_{22}c + 3\beta_{32}c^2$

$2\beta_{21} + 6\beta_{31}c = 2\beta_{22} + 6\beta_{32}c$

Spline Basis Functions

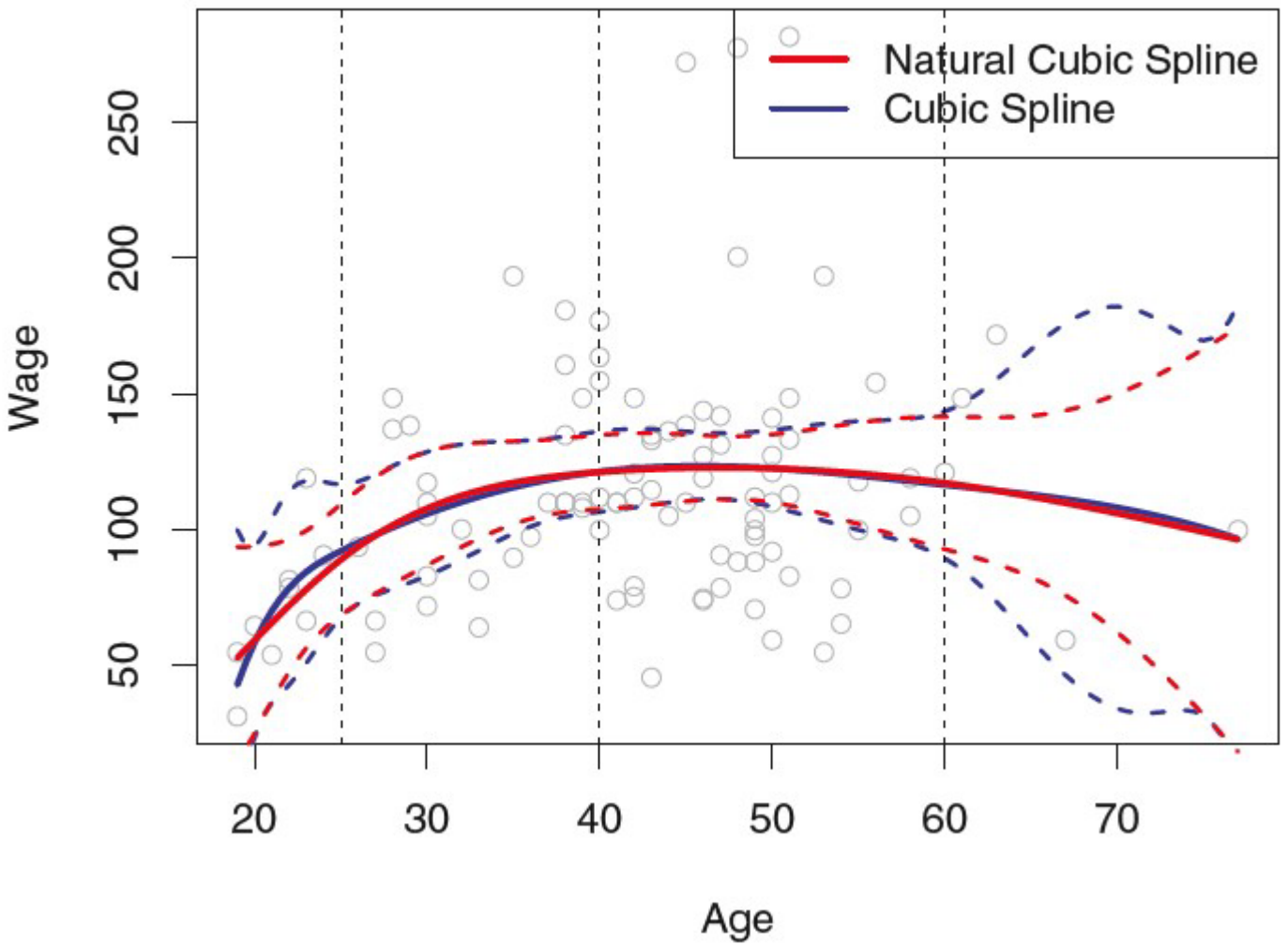
- A degree-*d* spline is a piecewise degree-*d* polynomial with continuity up to the *d*-1*th* derivative.
- A cubic spline with *K* knots can be modelled with basis functions as, $y_i = \beta_0 + \beta_1 b_1(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$
- One set of basis functions for the cubic splines are, $b_i(x) = x, x^2,$ and x^3. Then add one truncated power function for each knot,

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

  where ξ is a knot. By adding these terms there will be a discontinuity in the third derivative only.
- Thus, we would do linear regression on a model with an intercept and *K*+3 predictors, $X, X^2, X^3, h(x,\xi_1) \ldots h(x,\xi_K)$ and hence *K*+4 degrees of freedom.

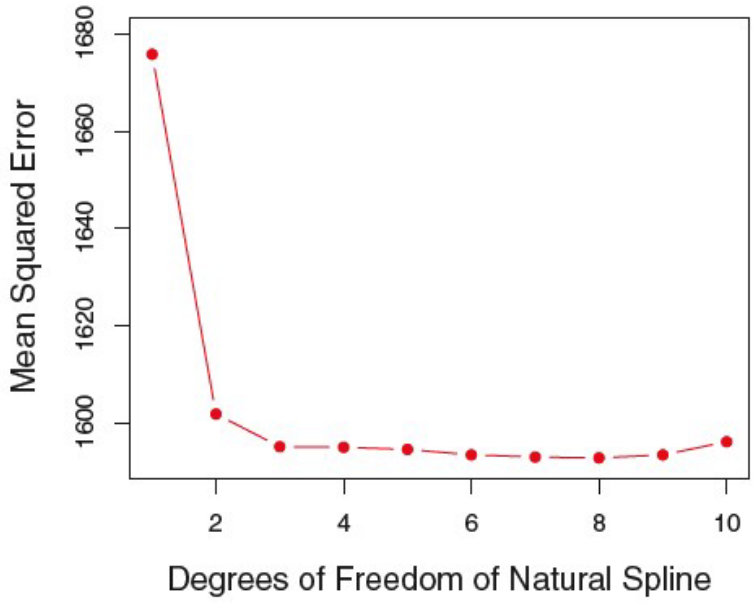
Natural Cubic Spline

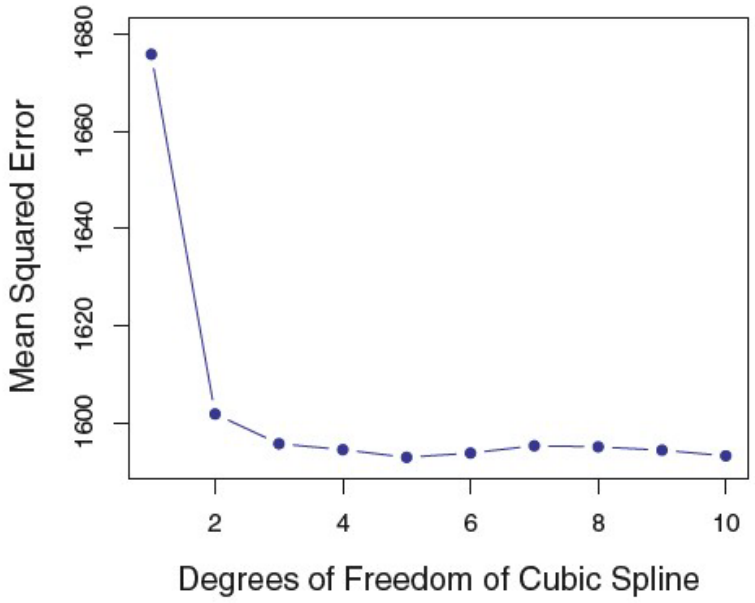
- The cubic spline can have very high variance at the boundaries. A natural spline is a regression spline with an additional boundary constraint. At the boundary when *x* is less than the first knot or greater than the last knot the function is linear.



Choosing Knots

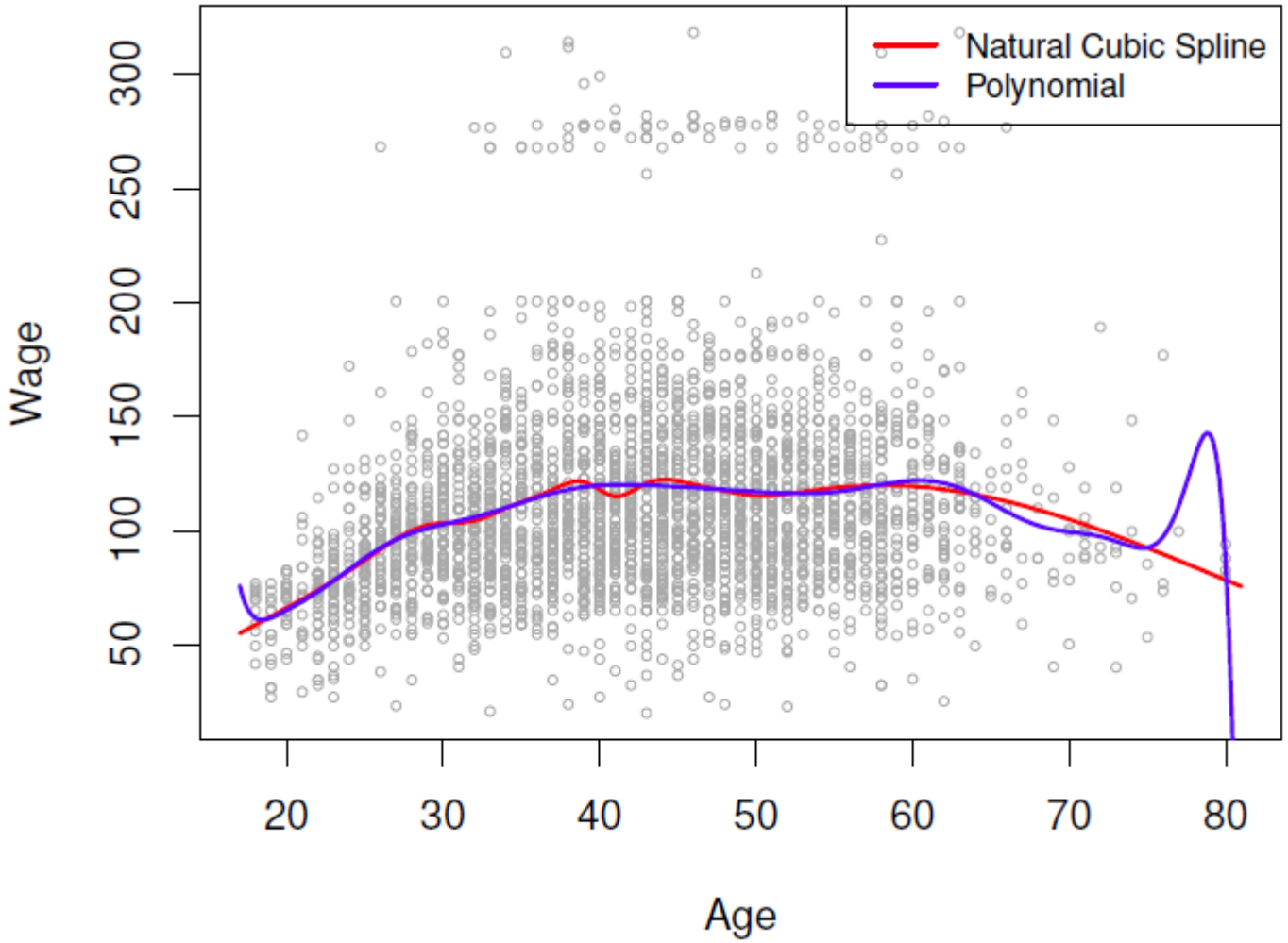
- Unlike FLAM the location of knots are not adaptively chosen. Usually, the number of degrees of freedom are chosen first.
- Knots may the be placed at uniform intervals or uniform quantiles of data.
- Cross-validation can also be used to chose the best number of degrees of freedom.
- With many predictors it may be easiest to just set the degrees of freedom to a constant number.
- On the right is the wage data. 10 fold CV.





Splines vs polynomials

- A 15th order polynomial vs. a cubic spline with 15 degrees of freedom.
- The high order features, e.g. x^{15}, make the behavior of the polynomial unpredictable especially at the boundary.



Smoothing Splines

- A regression function, $g(x_i)$, may be made sufficiently complex that the RSS is close to 0 or actually zero, but this will result in overfitting and a model which is very jumpy since it is chasing every observation.
- It is typically preferable to have a small RSS and a model which is smooth.
- A smoothing spline accomplishes by minimizing the following objective function, $\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$
- This function has the standard RSS loss function and a penalty function which integrates the regression functions second derivative over the predictors range.
- The second derivative measures the amount by which the slope is changing. The second derivative of a straight line is 0.

Smoothing Splines

- As λ approaches 0 we just get the least squares estimate.
- As $\lambda \to \infty$ g become a straight line (since the second derivative of a linear function is zero).
- The function that minimizes the smoothing spline objective function is a piecewise cubic polynomial with knots at unique values $x_1,\ldots,x_n$ with continuous first and second derivatives and straight lines at the end regions, e.g a natural cubic spline.
- It is not the same as the standard natural spline but shrunken due to $\lambda > 0$.

Choosing the Smoothing Parameter

- λ controls the smoothness and hence the effective degrees of freedom (df_λ). When λ=0 then df_λ can be n and at $\lambda = \infty$, df_λ =2.
- Although the smoothing spline has n-parameters and hence n degrees of freedom they are heavily constrained and shrunken. Thus, the effective degrees of freedom reflect the flexibility of the model.
- High df_λ mean more flexibility (low bias, high variance), low df_λ just the opposite.
- We can find the df_λ from the equation, $\hat{g}_\lambda = \boldsymbol{S}_\lambda \boldsymbol{y}$, where $\hat{g}_\lambda$ is the solution to the smoothing spline objective function, e.g. the fitted values at the training points x_1,…, x_n. Then $df_\lambda = tr(\boldsymbol{S}_\lambda)$.
- Computational details: $\boldsymbol{S}_\lambda = \boldsymbol{N}(\boldsymbol{N}^T\boldsymbol{N} + \lambda\boldsymbol{\Omega}_N)^{-1}$, where $\boldsymbol{N}$ is the $n \times n$ matrix of basis functions for the n-observations and $\{\boldsymbol{\Omega}_N\}_{jk} = \int N_j''(t)N_k''(t)\,dt$.

Choosing the Smoothing Parameter

- We can choose the best λ using cross-validation.
- There is a simple calculation for a leave-one-out cross validation error which is,

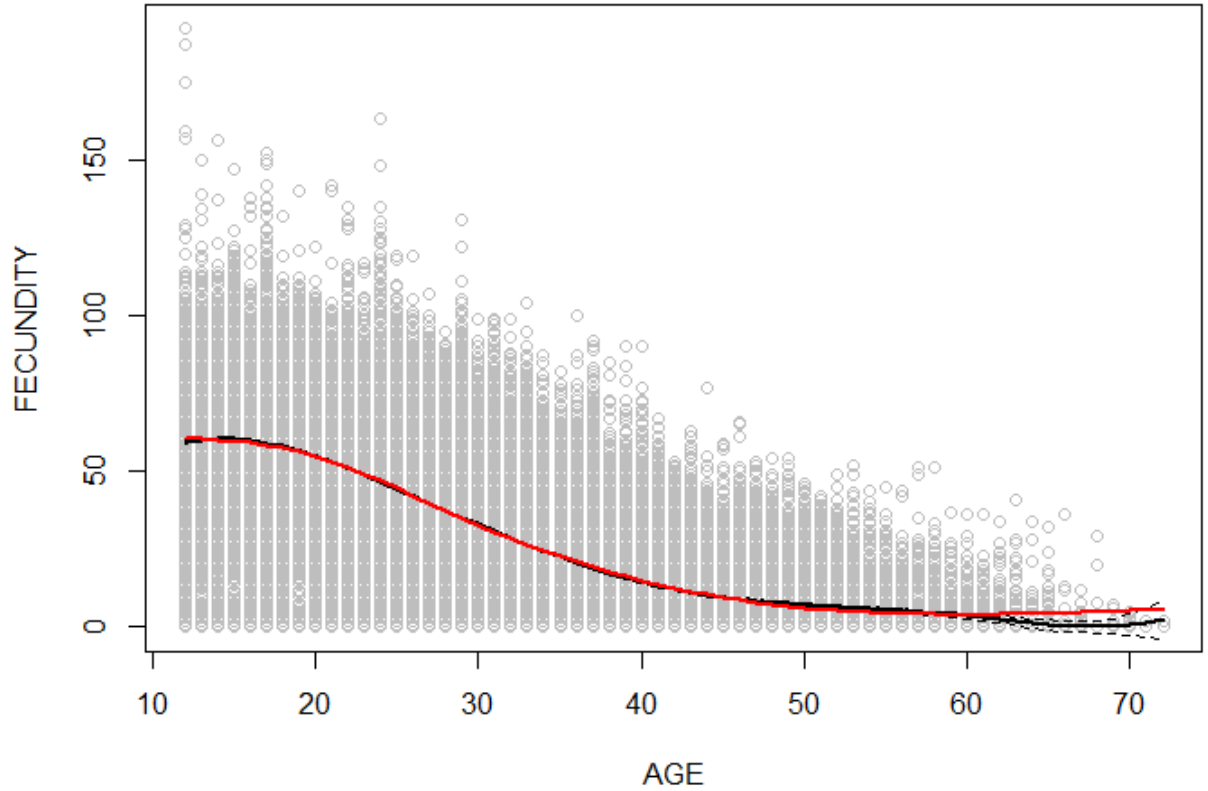
$$RSS_{cv}(\lambda) = \sum_{i=1}^{n} \left(y_i - \hat{g}_{\lambda}^{(-i)}(x_i) \right)^2 = \sum_{i=1}^{n} \left[\frac{y_i - \hat{g}_{\lambda}(x_i)}{1 - \{\boldsymbol{S}_{\lambda}\}_{ii}} \right]^2,$$

  where $\hat{g}_{\lambda}^{(-i)}(x_i)$ is the fitted (leaving observation-*i* out) function evaluated at x_i.
- This cross-validation can be estimated from just a single fit of the data as suggested by the right hand side of the equation above.

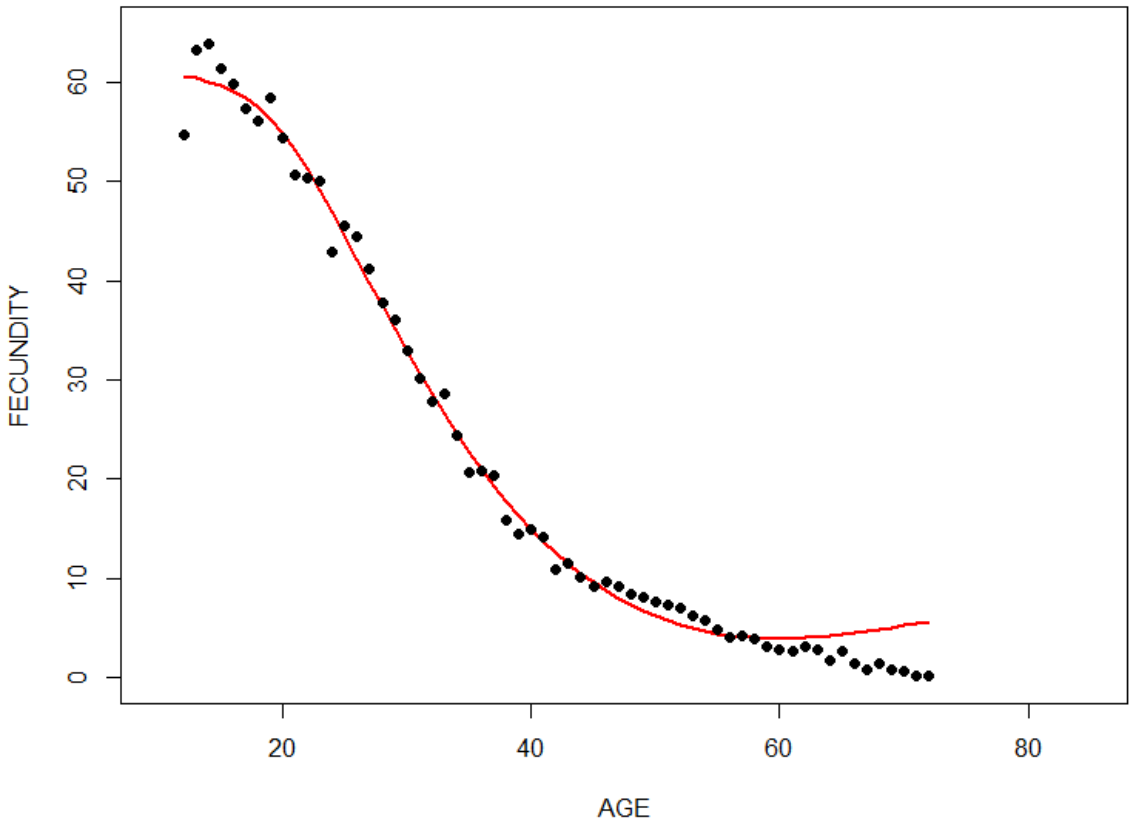
Example: age-specific fecundity, cubic spline, natural spline

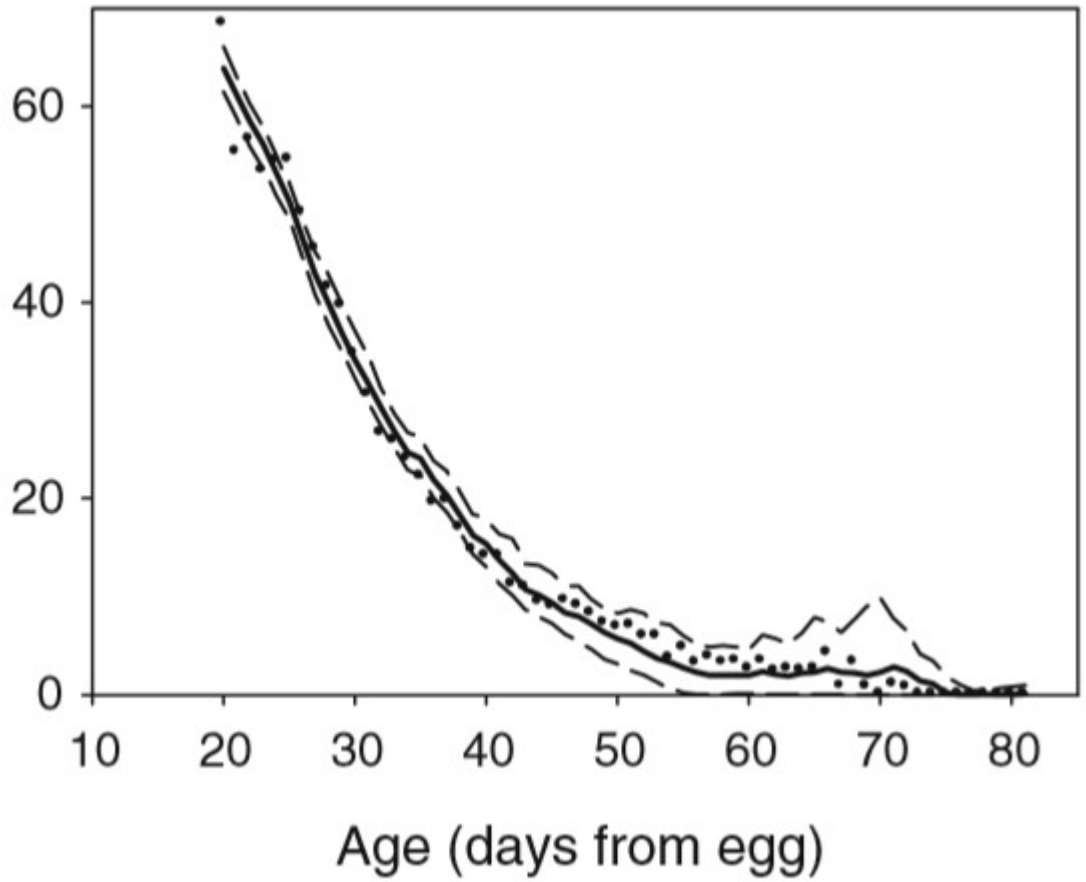
```
library(splines)
# Generate cubic splines at three knots,df=3+K=6 (since the default is no intercept)
fit<-lm(Fecundity~bs(Age ,knots =c(25 ,40 ,60),degree=3),data=fec.age)#bs is the B-spline
                                                                      function

age.grid<- seq(from=12,to=72)
pred=predict (fit ,newdata =list(Age =age.grid),se=T)
# Plot the raw data, spline fit +- 2s.e.'s
plot(fec.age$Age,fec.age$Fecundity,col="gray",xlab="AGE",ylab="FECUNDITY")
lines(age.grid,pred$fit,lwd =2)
lines(age.grid,pred$fit+2*pred$se,lty ="dashed")
lines(age.grid,pred$fit-2*pred$se,lty ="dashed")

# Let R choose the knots
attr(bs(fec.age$Age,df=6),"knots")
25% 50% 75%
 19  28  37
# Natural Spline
fit2<- lm(Fecundity~ns(Age,df=4),data=fec.age)
#ns is the natural spline function. Default is
# no intercept so there are df-1 knots
pred2<- predict(fit2,newdata=list(Age=age.grid),se=T)
lines(age.grid,pred2$fit,col="red",lwd=2)
```

RAW DATA: N=87,149



Compare Natural Spline to Evolutionary Model

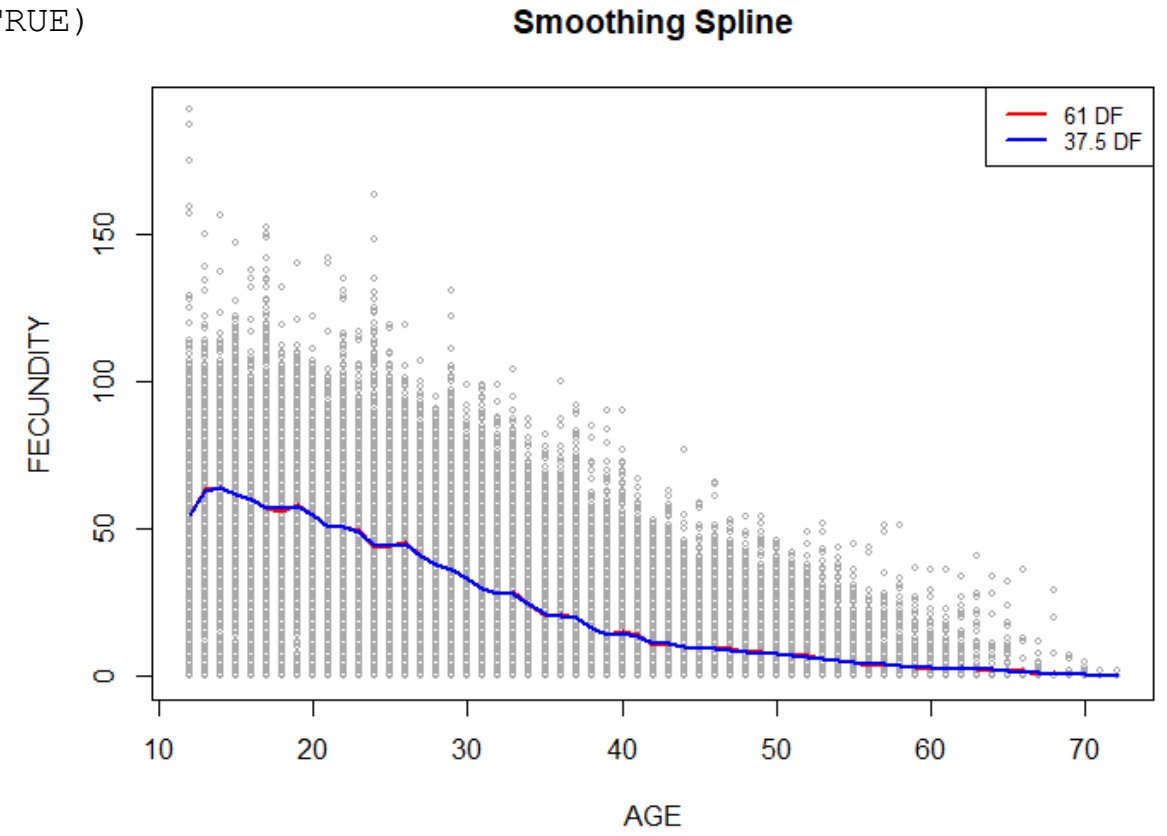




$CO_{1\text{-}3}$ with mean fecundity and fitted model and confidence intervals.
From Mueller et al. (2007) *Biogerontology* **8**: 147-161. The natural spline model has 4 degrees of freedom and the evolutionary model has 4 parameters or 4 degrees of freedom also.

Example: age-specific fecundity, smoothing spline

```
# Smoothing Spline
plot(fec.age$Age,fec.age$Fecundity,cex=0.5,col="darkgrey",xlab="AGE",ylab="FECUNDITY")
title("Smoothing Spline")
# First we specify that there will 61 degrees if freedom in "fit3", corresponding to 61 unique ages.
fit3<- smooth.spline(fec.age$Age,fec.age$Fecundity,df=61,cv=TRUE)
# "fit4" will use the leave-one-out CV error to choose the best λ
fit4<- smooth.spline(fec.age$Age,fec.age$Fecundity,cv=TRUE)
fit4$df #effective degrees of freedom
[1] 37.4882
lines(fit3,col="red",lwd=2)
lines(fit4,col="blue",lwd=2)
legend("topright",legend=c("61 DF","37.5 DF"),
col=c("red","blue"),lty=1,lwd=2,cex=0.8)
# The CV error should be smaller in "fit4" compared
# to "fit3"
> fit4$cv.crit
[1] 392.5167
> fit3$cv.crit
[1] 392.4659
# This difference is probably attributable to
# different random folds used in fit3 and fit4.
```



Local Regression

- Fit linear models only to target points using nearby observations.

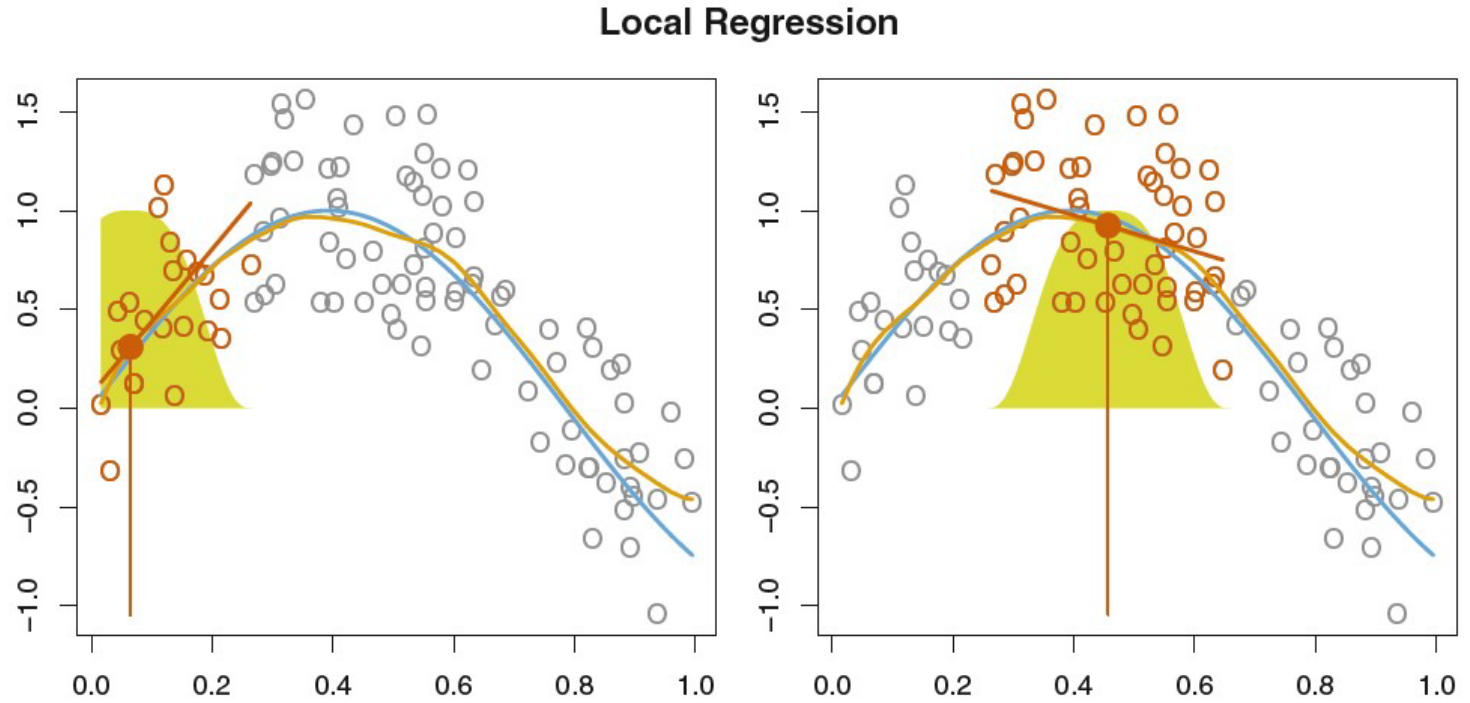


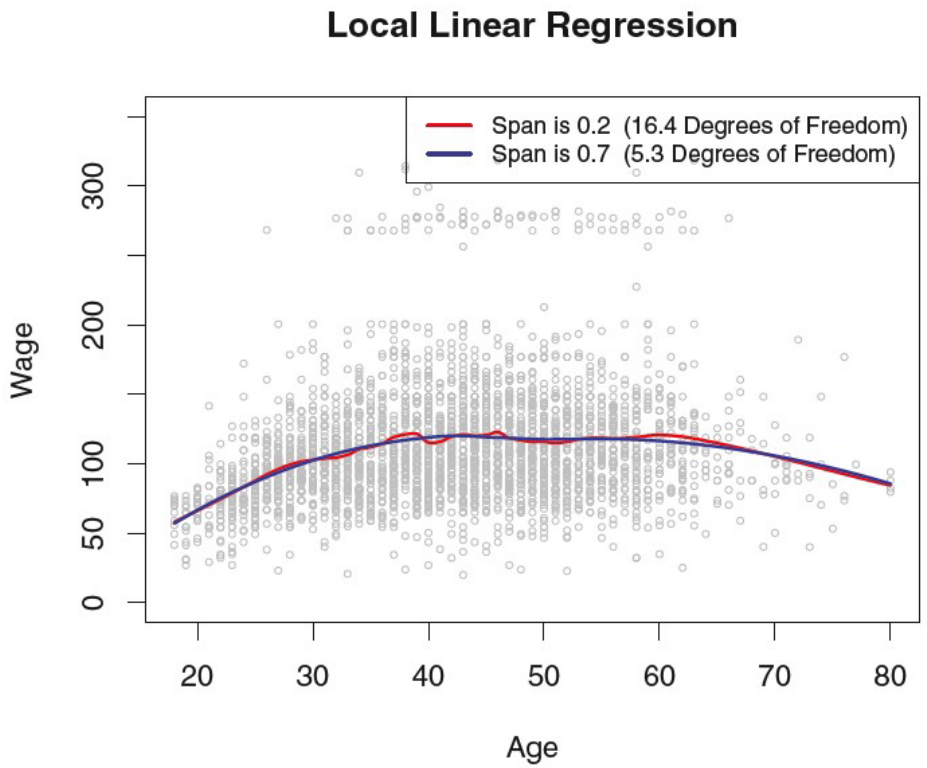
FIGURE 7.9. *Local regression illustrated on some simulated data, where the blue curve represents $f(x)$ from which the data were generated, and the light orange curve corresponds to the local regression estimate $\hat{f}(x)$. The orange colored points are local to the target point x_0, represented by the orange vertical line. The yellow bell-shape superimposed on the plot indicates weights assigned to each point, decreasing to zero with distance from the target point. The fit $\hat{f}(x_0)$ at x_0 is obtained by fitting a weighted linear regression (orange line segment), and using the fitted value at x_0 (orange solid dot) as the estimate $\hat{f}(x_0)$.*

Local Regression

- Algorithm
  1. Gather the fraction $s=k/n$ of training points whose x_i are closest to x_0.
  2. Assign a weight $K_{i0}=K(x_i, x_0)$ to each point in this neighborhood, so that the point furthest from x_0 has a weight zero, and the closest has the highest weight. All but these k nearest neighbors get weight zero.
  3. Fit a weighted least squares regression of the y_i on the x_i using the aforementioned weights, by finding $\hat{\beta}_0$ and $\hat{\beta}_1$ that minimize, $\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$.
  4. The fitted value at x_0 is given by, $\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$

Local Regression

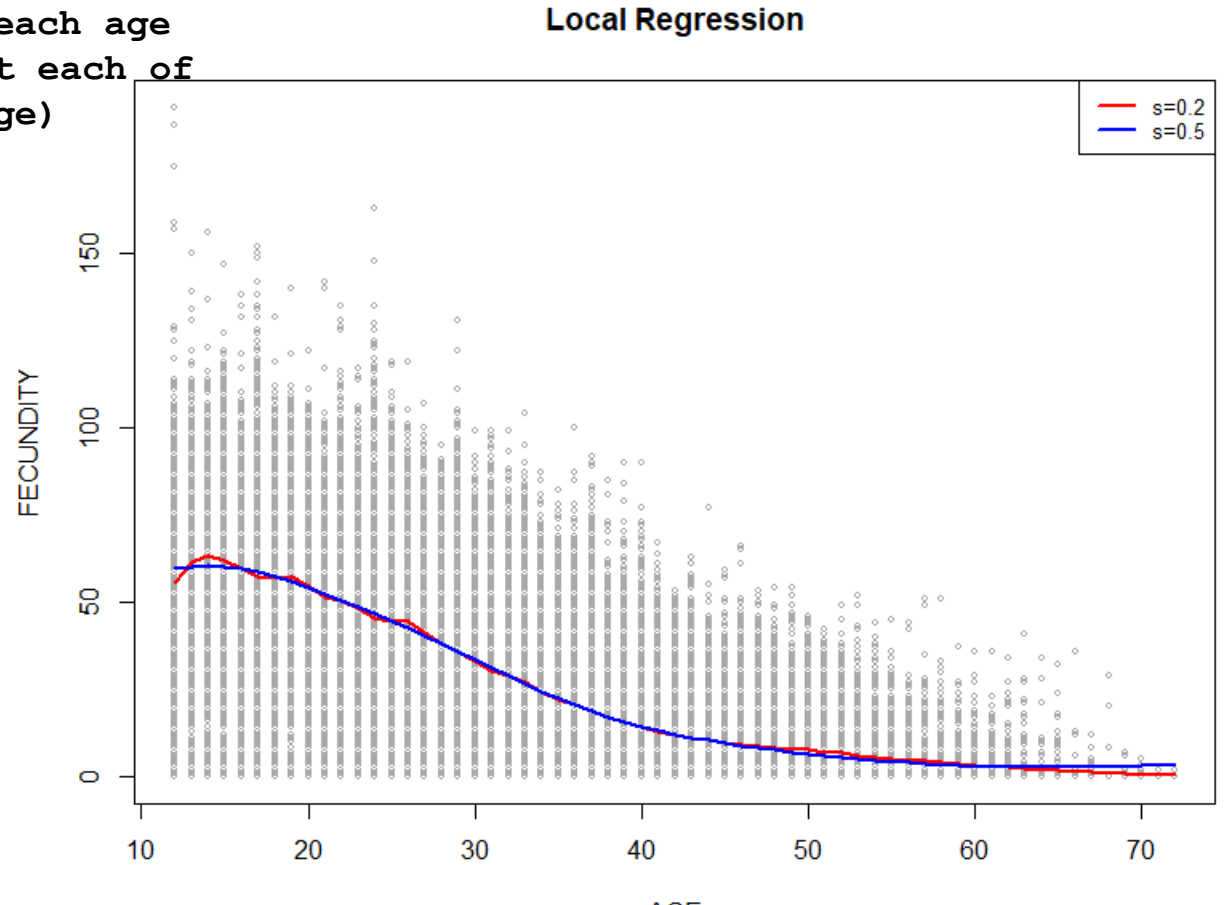
- The local regression with more degrees of freedom tends to be jumpy.
- Local regression in more than 3 or 4 dimensions will quickly run out of data as we saw with the curse of dimensionality.



Example: Local Regression

```
# Local Regression
plot(fec.age$Age,fec.age$Fecundity,cex=0.5,col="darkgrey",xlab="AGE",ylab="FECUNDITY")
title("Local Regression")
fit5<- loess(fec.age$Fecundity~fec.age$Age,span=.2,data=fec.age)# span controls the proportion of points
fit6<- loess(fec.age$Fecundity~fec.age$Age,span=.5,data=fec.age)# to be used in the local regression
                                                               # degree defaults to 2 a quadratic fit

# The predict function doesn't work as expected. This may be
# due to the large number of replicate observations at each age
# The next line identifies the first row in fec.age that each of
# the unique ages appear, e.g. uages<- unique(fec.age$Age)
unique.locations<- sapply(1:length(uages),function(x)
  which(fec.age[,1]==uages[x])[1])
lines(uages,fit5$fitted[unique.locations],
  col="red",lwd =2)
lines(uages,fit6$fitted[unique.locations],
  col="blue",lwd =2)
legend("topright",legend=c("s=0.2","s=0.5"),
  col=c("red","blue"),lty=1,lwd=2,cex=0.8)
```

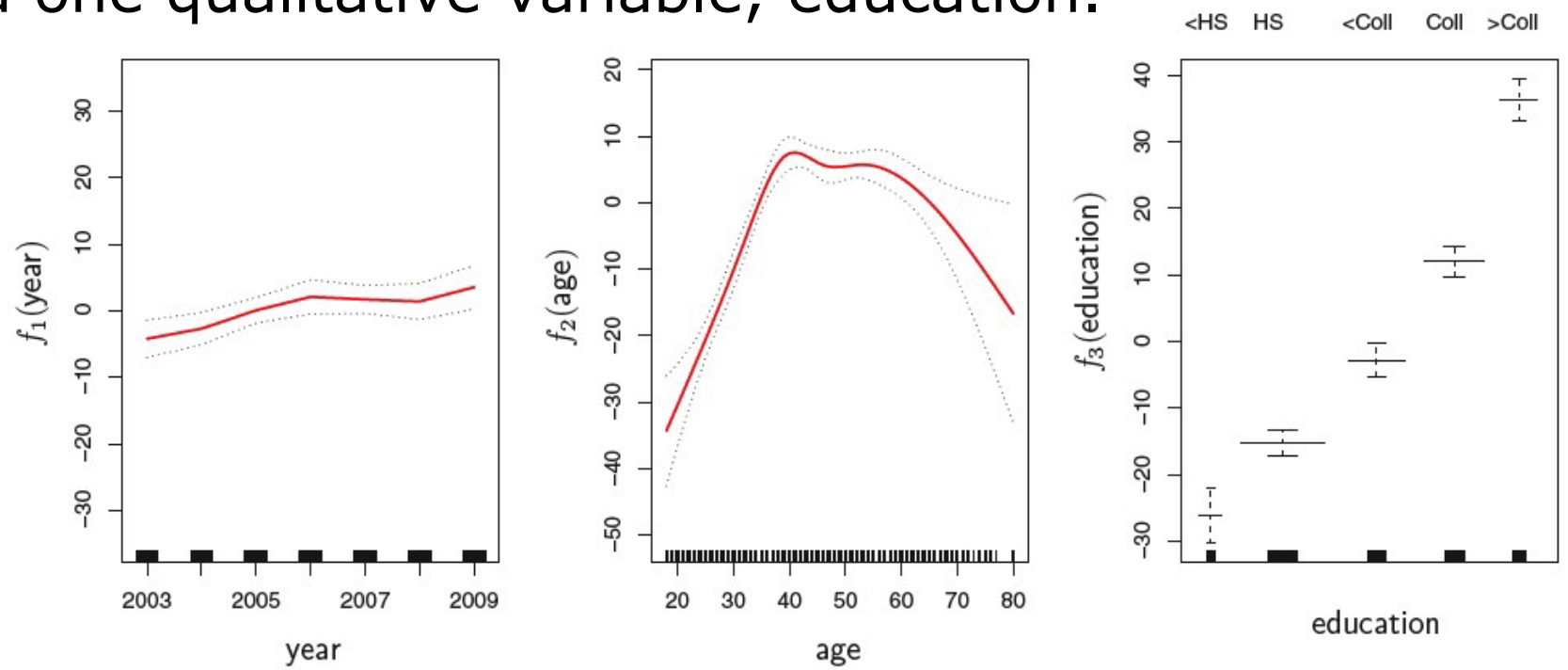
fit5$fitted is a vector with the predicted fecundity of all 87,149 observations in the same order as the original data file – fec.age.



Generalized Additive Models

- The methods considered to this point can now be used to model responses with multiple predictors.
- The generalized additive model (GAM) adds a different functional form for each predictor, $y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \epsilon_i$
- For the wage data we can include two quantitative variables, year and age, and one qualitative variable, education.

The first two functions are natural splines and the last function is a step function.



Fitting Additive Models

- This material comes from the Elements book so we will change notation.
- The general model is, $Y = \alpha + \sum_{j=1}^{p} f_j(X_j) + \epsilon$. Each function will be modelled as a penalized residual sum of squares (PRSS), as we used with smoothing splines.
- We seek to minimize,

$$PRSS(\alpha, f_1, \ldots, f_p) = \sum_{i=1}^{N} \left(y_i - \alpha - \sum_{j=1}^{p} f_j(x_{ij}) \right)^2 + \sum_{j=1}^{p} \lambda_j \int f_j''(t_j)^2 dt_j$$

- A cubic spline for each f_j will be a minimizer to this objective function with knots at the unique values of x_{ij}, i=1,…,N.
- To guarantee a unique solution we require $\sum_{i=1}^{N} f_j(x_{ij}) = 0$ for all j (the functions average 0 over all data). In addition, the matrix of input values (x_{ij}) must have full column rank, e.g. no columns can not be linear combinations of each other.

Fitting Additive Models Algorithm

- 1. Initialize: $\hat{\alpha} = \frac{1}{N}\sum_1^N y_i, \hat{f}_j \equiv 0, \forall i, j.$

  2. Cycle: j=1, 2, …,p,…, 1, 2, .., p,…,

  $\hat{f}_j \leftarrow S_j\left[\{y_i - \hat{\alpha} - \sum_{k\neq j}\hat{f}_k(x_{ik})\}_1^N\right]$

  $\hat{f}_j \leftarrow \hat{f}_j - \frac{1}{N}\sum_{i=1}^N \hat{f}_j\left(x_{ij}\right)$

  until the functions $\hat{f}_j$ change less than a prespecified threshold.
- S_j means fit to a smoothing cubic spline to the residuals between {}. This smoothing spline is the new estimate, $\hat{f}_j$. After this is done the functions are updated before the next function is fit to the residuals.
- The technique is called “backfitting”.
- The second step in 2 should not be needed since the smoothing spline should have a zero mean. But machine rounding can cause slippage, so this is an advisable numerical technique.

GAM pros and cons

- Allow for flexible, non-linear fits.
- Each predictor can be examined for its effects on the outcome holding the other predictors fixed.
- Although predictors must be additive, interactions can be added as a separate additive factor.
- Can be thought of as a compromise between a linear model and a fully non-parametric model – like random forests.