# 21) Regression Splines, Smoothing Splines, Local Regression

Vitor Kamada

March 2018

# Reference

Tables, Graphics, and Figures from

**An Introduction to Statistical Learning**

James et al. (2017): Chapters: 7.4, 7.5, 7.6, and 7.8.2

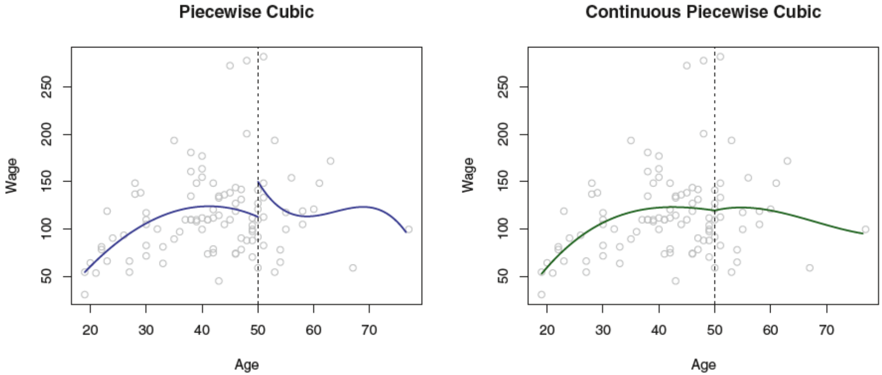
# Piecewise Polynomials

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & (I) \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & (II) \end{cases}$$

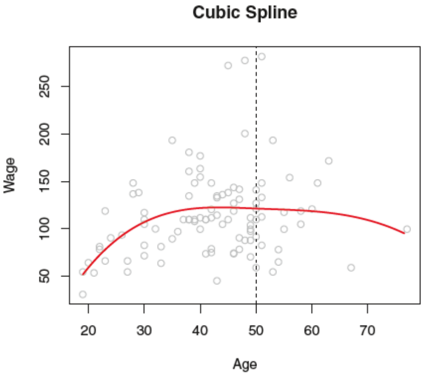
$$(I) \text{ } if \text{ } x_i < c$$

$$(II) \text{ } if \text{ } x_i \geq c$$

# Unconstrained vs Constrained



# Continuous, Continuous First and Second Derivatives vs Linear Continuous



# Cubic Spline Basis Representation

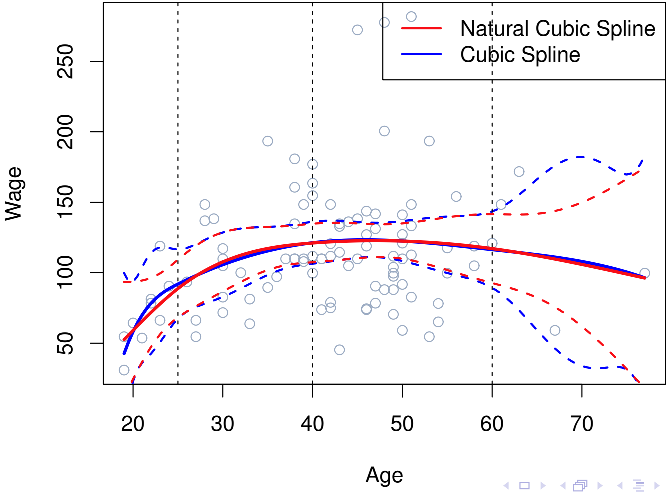
$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \ldots + \beta_{k+3} b_{k+3}(x_i) + \epsilon_i$$

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

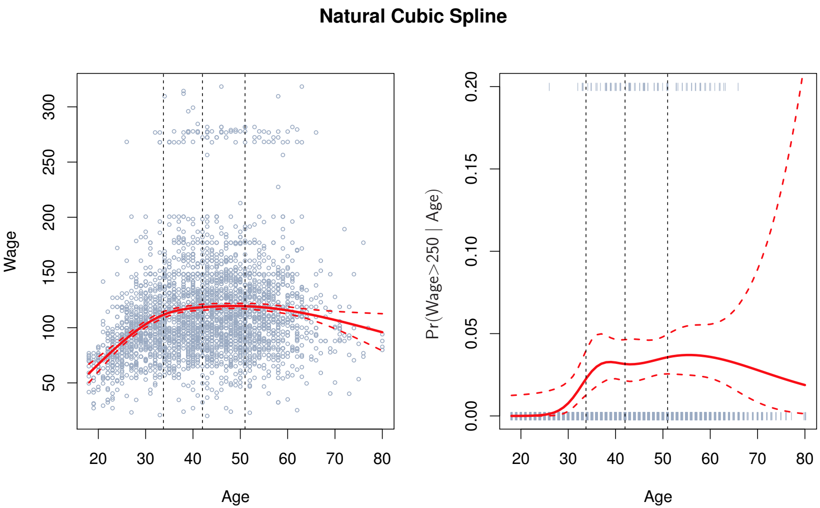
Intercept and 3+K Predictors:

$$1, X, X^2, X^3, h(x,\xi_1), h(x,\xi_2), \ldots, h(x,\xi_K)$$

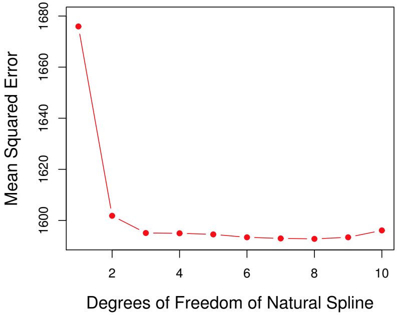
# Three Knots

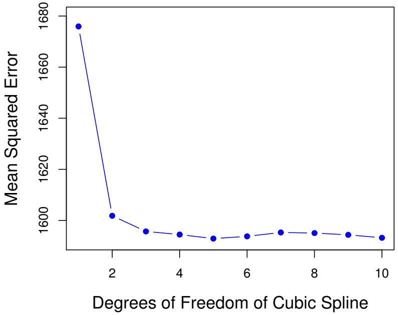

Natural Cubic Spline
Cubic Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70

# Spline with Three Knots (25th, 50th, and 75th) vs Logistic Regression

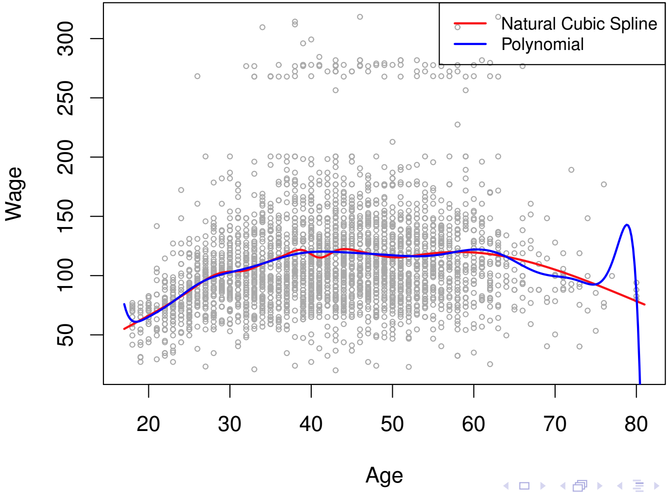


# Ten-fold Cross-Validation



# Spline with 15 df vs Degree-15 Polynomial



# Smoothing Splines

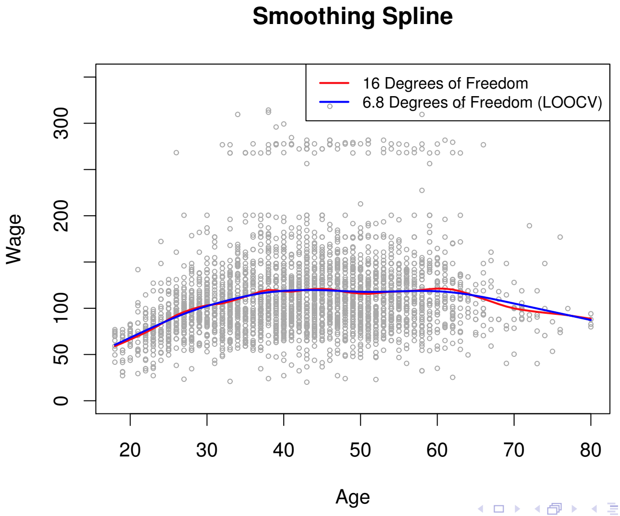
$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

$$\hat{g}_\lambda = S_\lambda y, \qquad df_\lambda = \sum_{i=1}^{n}\{S_\lambda\}_{ii}$$

$$RSS_{cv}(\lambda) = \sum_{i=1}^{n}(y_i - \hat{g}_\lambda^{(-i)}(x_i))^2$$

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

# 16 Effective df vs 6.8 Effective df Resulted by Leave-One-Out Cross-Validation



# library(ISLR); attach(Wage)

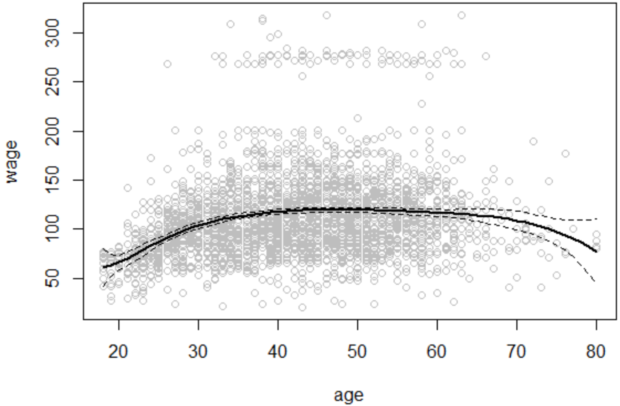
```
agelims=range(age)
age.grid=seq(from=agelims[1],to=agelims[2])

library(splines)
fit=lm(wage~bs(age,knots=c(25,40,60)),data=Wage)
pred=predict(fit,newdata=list(age=age.grid),se=T)
```

# plot(age,wage,col="gray");

lines(age.grid,pred$fit,lwd=2)

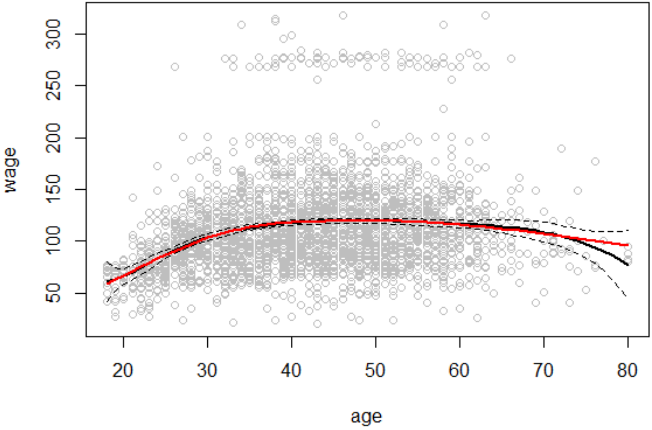
lines(age.grid,pred$fit+2*pred$se,lty="dashed")

lines(age.grid,pred$fit-2*pred$se,lty="dashed")



# fit2=lm(wage~ns(age,df=4),data=Wage)

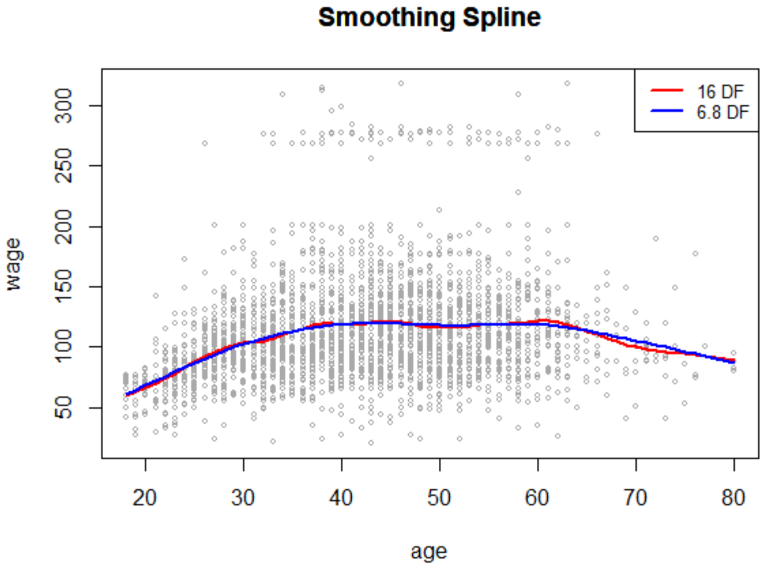
```
pred2=predict(fit2,newdata=list(age=age.grid),se=T)
lines(age.grid, pred2$fit,col="red",lwd=2)
```



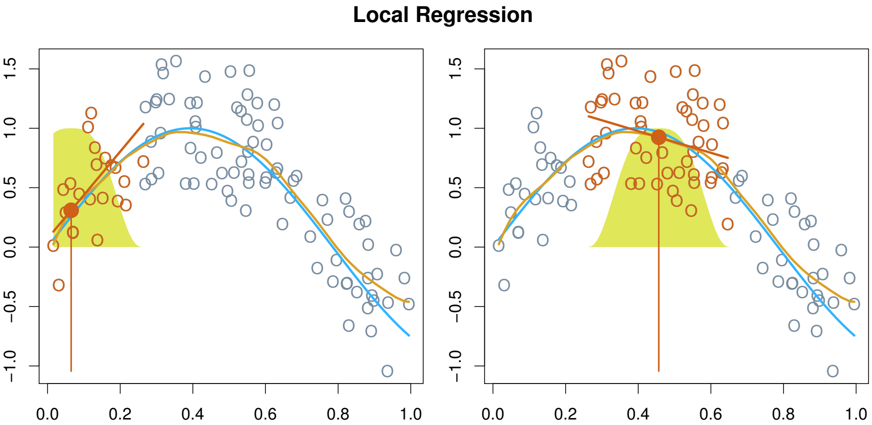
# plot(age,wage,xlim=agelims,cex=.5,col="darkgrey")

```
title("Smoothing Spline")
fit=smooth.spline(age,wage,df=16)
fit2=smooth.spline(age,wage,cv=TRUE)
fit2$df
lines(fit,col="red",lwd=2)
lines(fit2,col="blue",lwd=2)
legend("topright",legend=c("16 DF", "6.8 DF"),
col=c("red","blue"), lty=1,lwd=2,cex=.8)
```

# df =16 vs Cross-Validation



# Local Regression - Simulated Data



# Local Regression at $X = x_0$

1. Gather $s = \frac{k}{n}$ training points whose $x_i$ are closer to $x_0$
2. Assign a weight $K_{i0} = K(x_i, x_0)$
3. Fit $\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$
4. Fitted value at $x_0$ is $\hat{f}(x_0) = \hat{\beta}_0 - \hat{\beta}_1 x_0$

# plot(age,wage,xlim=agelims,cex=.5,col="darkgrey")

```
title("Local Regression")
fit=loess(wage~age,span=.2,data=Wage)
fit2=loess(wage~age,span=.5,data=Wage)
lines(age.grid,predict(fit,data.frame(age=age.grid)),
col="red",lwd=2)
lines(age.grid,predict(fit2,data.frame(age=age.grid)),
col="blue",lwd=2)
legend("topright",legend=c("Span=0.2","Span=0.5"),
col=c("red","blue"),lty=1,lwd=2,cex=.8)
```

# Neighborhood of 20% vs 50% of the Observations