ISL - Chapter 7 Lab Tutorials
Moving Beyond Linearity

An introduction to Statistical Learning, with Applications in R
- G. James, D. Witten, T. Hastie, R. Tibshirani

Thu Nguyen

09 July, 2019

Contents

Main Contents:

1. Polynomial Regression
2. Step Functions
3. Basis Functions
4. Regression Splines
5. Smoothing Splines
6. Local Regression
7. Generalized Additive Models

```r
library(ISLR)
attach(Wage)
```

7.8. Lab: Non-linear Modeling

7.8.1. Polynomial Regression and Step Functions

Polynomial Regression

To specify a polynomial such as $f(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4$, in `R`: `poly(x,4)`:

```
fit <- lm(wage ~ poly(age,4), data = Wage)
coef(summary(fit))
```

```
##                  Estimate Std. Error    t value                                Pr(>|t|)
## (Intercept)     111.70361  0.7287409 153.283015 0.0000000000000000000000000000000000000
## poly(age, 4)1   447.06785 39.9147851  11.200558 0.0000000000000000000000000014846042765
## poly(age, 4)2  -478.31581 39.9147851 -11.983424 0.0000000000000000000000000000002355831
## poly(age, 4)3   125.52169 39.9147851   3.144742 0.00167862178126833294644626448643975891
## poly(age, 4)4   -77.91118 39.9147851  -1.951938 0.05103864982784254988867900237892172299
```

Other equivalent ways to specify a polynomial in `R`:

```
fit2 <- lm(wage ~ poly(age,4, raw = T), data = Wage)
coef(summary(fit2))
```

```
##                            Estimate     Std. Error   t value     Pr(>|t|)
## (Intercept)           -184.1541797743 60.04037718327 -3.067172 0.0021802539
## poly(age, 4, raw = T)1   21.2455205321  5.88674824448  3.609042 0.0003123618
## poly(age, 4, raw = T)2   -0.5638593126  0.20610825640 -2.735743 0.0062606446
## poly(age, 4, raw = T)3    0.0068106877  0.00306593115  2.221409 0.0263977518
## poly(age, 4, raw = T)4   -0.0000320383  0.00001641359 -1.951938 0.0510386498
```

Explicitly specifying a polynomial using `I()` in the `formula`:

```
fit2a <- lm(wage ~ age + I(age^2) + I(age^3) + I(age^4), data = Wage)
coef(summary(fit2))
```

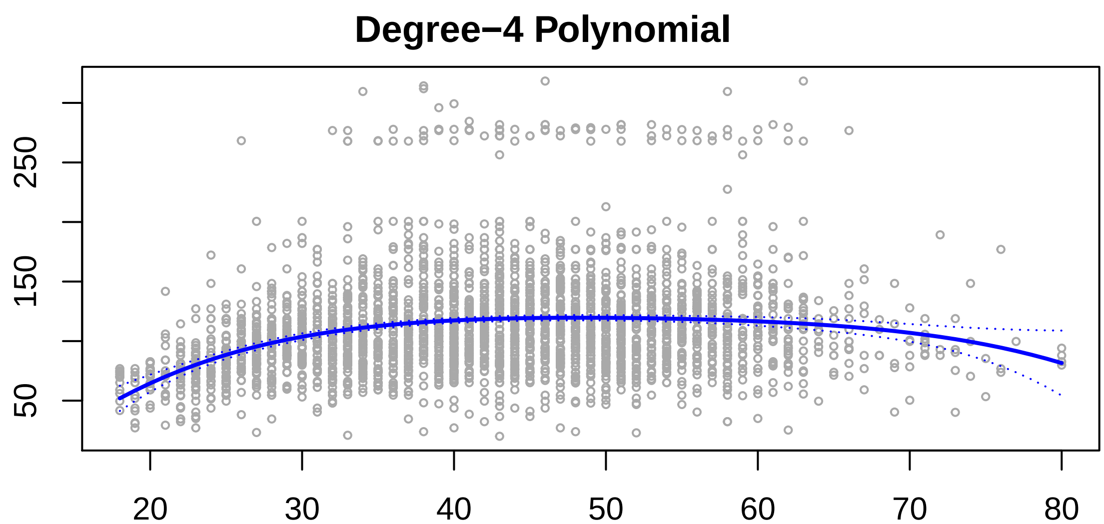
```
##                            Estimate     Std. Error   t value     Pr(>|t|)
## (Intercept)           -184.1541797743 60.04037718327 -3.067172 0.0021802539
## poly(age, 4, raw = T)1   21.2455205321  5.88674824448  3.609042 0.0003123618
## poly(age, 4, raw = T)2   -0.5638593126  0.20610825640 -2.735743 0.0062606446
## poly(age, 4, raw = T)3    0.0068106877  0.00306593115  2.221409 0.0263977518
## poly(age, 4, raw = T)4   -0.0000320383  0.00001641359 -1.951938 0.0510386498
```

Regression and Prediction:

```
agelims <- range(age)
age.grid <- seq(from = agelims[1], to = agelims[2])
preds <- predict(fit, newdata = list(age=age.grid), se=TRUE)
se.bands <- cbind(preds$fit + 2*preds$se.fit, preds$fit - 2*preds$se.fit)

par(mfrow = c(1,1), mar = c(3,3,0,.5), oma = c(0,0,2,0))
```

```r
plot(age, wage, xlim=agelims, cex=.5, col='darkgrey')
title('Degree-4 Polynomial', outer=T)
lines(age.grid, preds$fit, lwd=2, col='blue')             # Regression line
matlines(age.grid, se.bands, lwd=1, col='blue', lty=3)     # Standard Error line
```




```r
preds2 <- predict(fit2, newdata = list(age=age.grid), se=TRUE)
print(paste('Difference between with and without raw=T:', max(abs(preds$fit - preds2$fit))))
```

```
## [1] "Difference between with and without raw=T: 0.000000000078159700933611"
```

In performing a *polynomial regression*, the problem reduces to the degree of the polynomial, which can be approached by hypothesis tests.

```r
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 < 0.00000000000000022 ***
## 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
```

Interpretation:

- p-values of comparing Model 1 vs. Model 2 is practically 0 $\implies$ Model 1 is not sufficient and Model 2 is decidedly better
- similarly, between Models 2 and 3, Model 3 is superior
- between Models 4 and 5, p-value is .37 $\implies$ Model 5 is unnecessary
- at $p = 05$, either Models 3 or 4 is alright.

Alternatively, instead of `anova()`, p-value is already encoded in higher order polynomials:

```
coef(summary(fit.5))
```

```
##                  Estimate Std. Error      t value                                  Pr(>|t|)
## (Intercept)     111.70361  0.7287647  153.2780243 0.0000000000000000000000000000000000000
## poly(age, 5)1   447.06785 39.9160847   11.2001930 0.0000000000000000000000000014911107825
## poly(age, 5)2  -478.31581 39.9160847  -11.9830341 0.0000000000000000000000000000002367734
## poly(age, 5)3   125.52169 39.9160847    3.1446392 0.0016792128263079576268312909093083362 7
## poly(age, 5)4   -77.91118 39.9160847   -1.9518743 0.0510462313327174968535793198043393204 0
## poly(age, 5)5   -35.81289 39.9160847   -0.8972045 0.3696819659743993957690122442727442830 8
```

More elaborated models: $wage = f(\text{education}, p(\text{age}))$

```
fit.1 <- lm(wage ~ education + age, data = Wage)
fit.2 <- lm(wage ~ education + poly(age,2), data = Wage)
fit.3 <- lm(wage ~ education + poly(age,3), data = Wage)
anova(fit.1, fit.2, fit.3)
```

```
## Analysis of Variance Table
##
## Model 1: wage ~ education + age
## Model 2: wage ~ education + poly(age, 2)
## Model 3: wage ~ education + poly(age, 3)
##   Res.Df     RSS Df Sum of Sq        F              Pr(>F)
## 1   2994 3867992
## 2   2993 3725395  1    142597 114.6969 <0.0000000000000002 ***
## 3   2992 3719809  1      5587   4.4936              0.0341 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Problem: predicting if an individual eanrs more than $\$250,000$, which is *classification* problem. Before proceeding, we need to create a binary variable for `wage`, by the `I()` function:

```
fit <- glm(I(wage > 250) ~ poly(age,4), data = Wage, family = binomial)
preds <- predict(fit, newdata = list(age = age.grid), se=T)
```

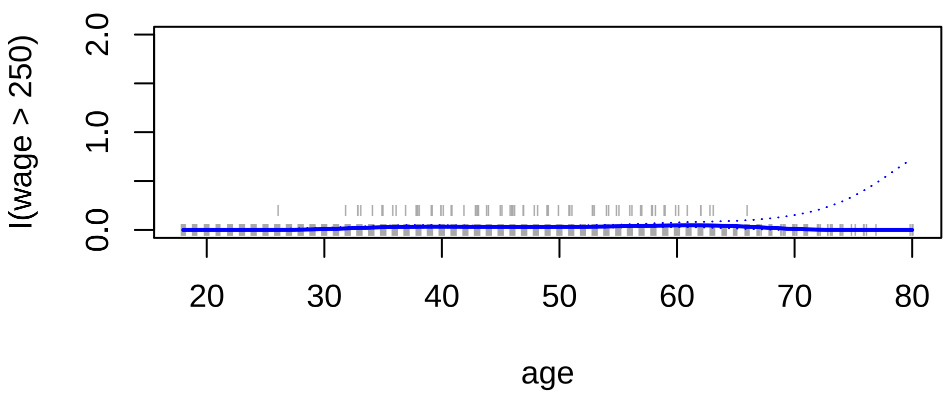
Recall the *logit* equation for *logistic regression*:

$$\log\left(\frac{\mathbb{P}(Y=1|X)}{1-\mathbb{P}(Y=1|X)}\right) = X\beta \quad \Longrightarrow \mathbb{P}(Y=1|X) = \frac{\exp(X\beta)}{1+\exp(X\beta)}$$

```
pfit <- exp(preds$fit) / (1 + exp(preds$fit))                    # transformation
se.bands.logit <- cbind(preds$fit + 2*preds$se.fit, preds$fit - 2*preds$se.fit)
se.bands <- exp(se.bands.logit) / (1 + exp(se.bands.logit))     # transformatino
preds <- predict(fit, newdata = list(age = age.grid), type = 'response', se=T)
```

When plotting, to prevent points close together from overlapping each other, use `jitter()`:

```
par(mar=c(4,4,0.5,0.5))
plot(age, I(wage > 250), xlim = agelims, type = 'n', ylim = c(0,2))
points(jitter(age), I((wage>250)/5), cex=.5, pch='l', col = 'darkgrey')
lines(age.grid, pfit, lwd = 2, col = 'blue')
matlines(age.grid, se.bands, lwd = 1, col = 'blue', lty = 3)
```



Step Functions

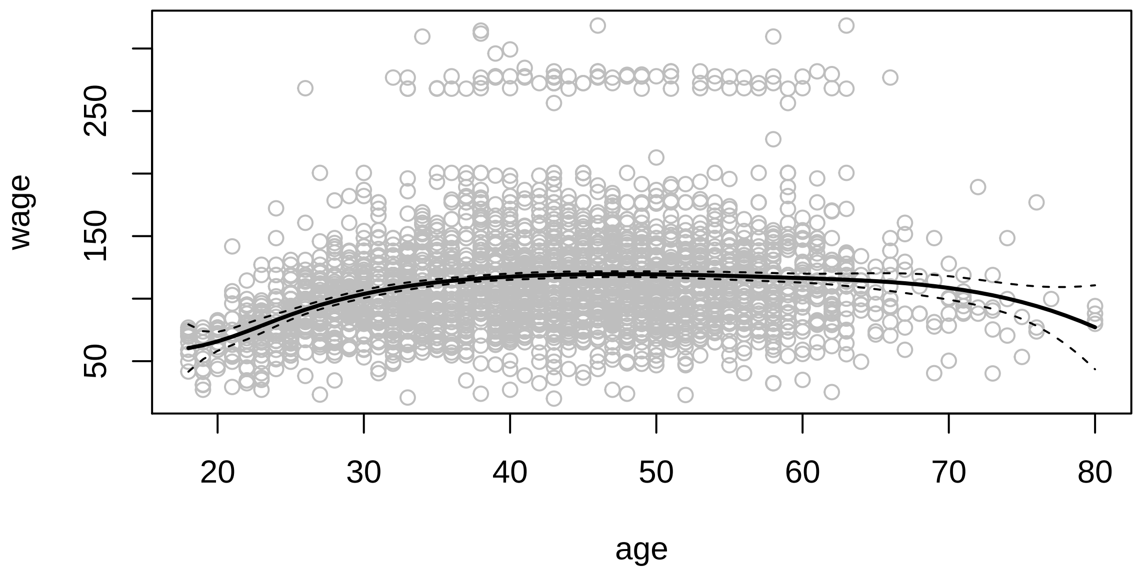
```
fit <- lm(wage ~ cut(age,4), data = Wage)      # cut(age,4) breaks age into 4 equal baskets
coef(summary(fit))
```

```
##                         Estimate Std. Error   t value                                      Pr(>|t|)
## (Intercept)            94.158392   1.476069 63.789970 0.0000000000000000000000000000000000000000000
## cut(age, 4)(33.5,49]   24.053491   1.829431 13.148074 0.0000000000000000000000000000000000001982315
## cut(age, 4)(49,64.5]   23.664559   2.067958 11.443444 0.0000000000000000000000000001040749536333972
## cut(age, 4)(64.5,80.1]  7.640592   4.987424  1.531972 0.1256350386559505760697419418647768907248973
```

7.8.2. Splines

To fit *regression splines*, we use `splines` package. To create a matrix of basis functions: `bs()`, within which, to specify `knots`: `knots = c()`. By default, *cubic splines* are created.

```
par(mar=c(4,4,.5,.5))
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')
```




```
print(paste('Degree of freedom from explicitly specified knots above:', dim(bs(age, knots = c(25,40,60)))[
```

```
## [1] "Degree of freedom from explicitly specified knots above: 6"
```

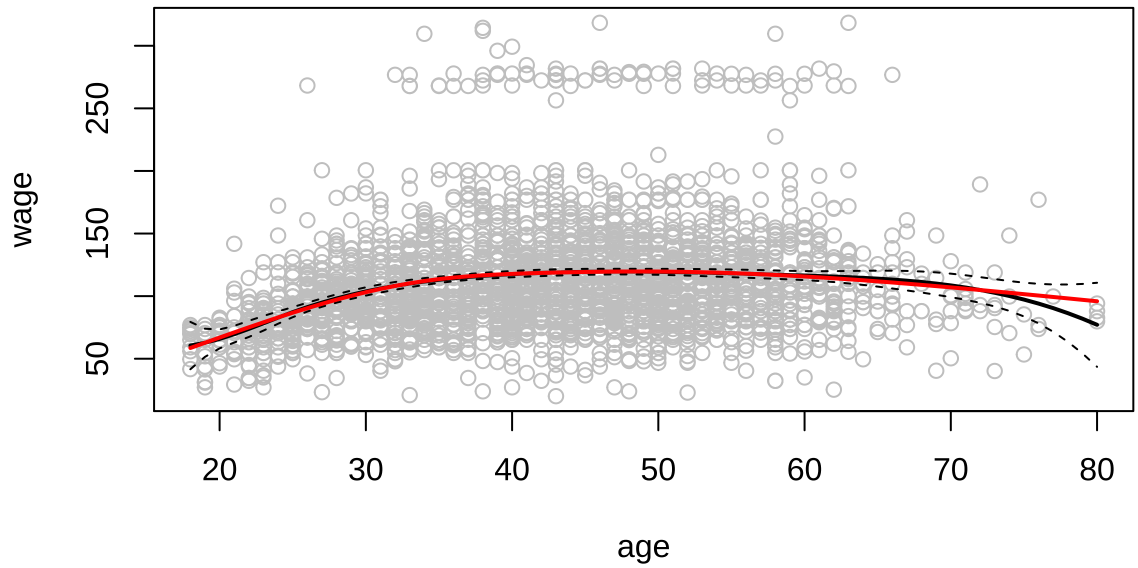
Alternatively, if $df = 6$ is specified instead of `knots`, `R` will choose the knots:

```
attr(bs(age, df = 6), 'knots')
```

```
##   25%   50%   75%
## 33.75 42.00 51.00
```

Alternatively, to fit a *natural spline*: `ns()`:

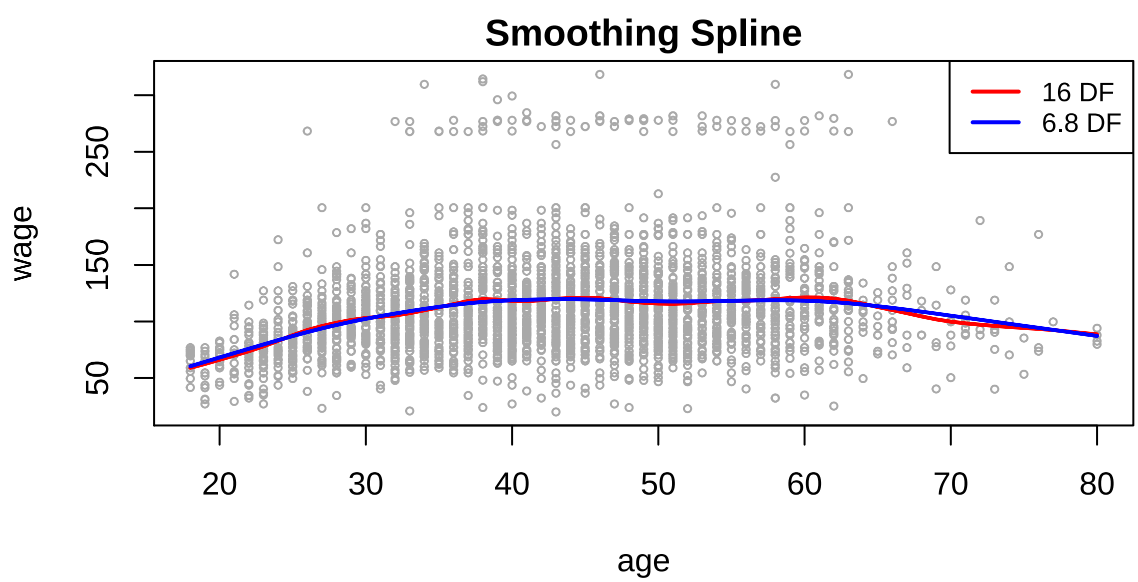
```
par(mar=c(4,4,.5,.5))
fit2 <- lm(wage ~ ns(age, df = 4), data = Wage)            # fit using natural spline, df = 4
pred2 <- predict(fit2, newdata = list(age = age.grid), se=T)
plot(age, wage, col = 'gray')
lines(age.grid, pred$fit, lwd = 2)
lines(age.grid, pred$fit + 2*pred$se, lty = 'dashed')
lines(age.grid, pred$fit - 2*pred$se, lty = 'dashed')
lines(age.grid, pred2$fit, col = 'red', lwd = 2)
```



Smoothing Spline

To fit a *smoothing spline*: we use `smooth.spline()`:

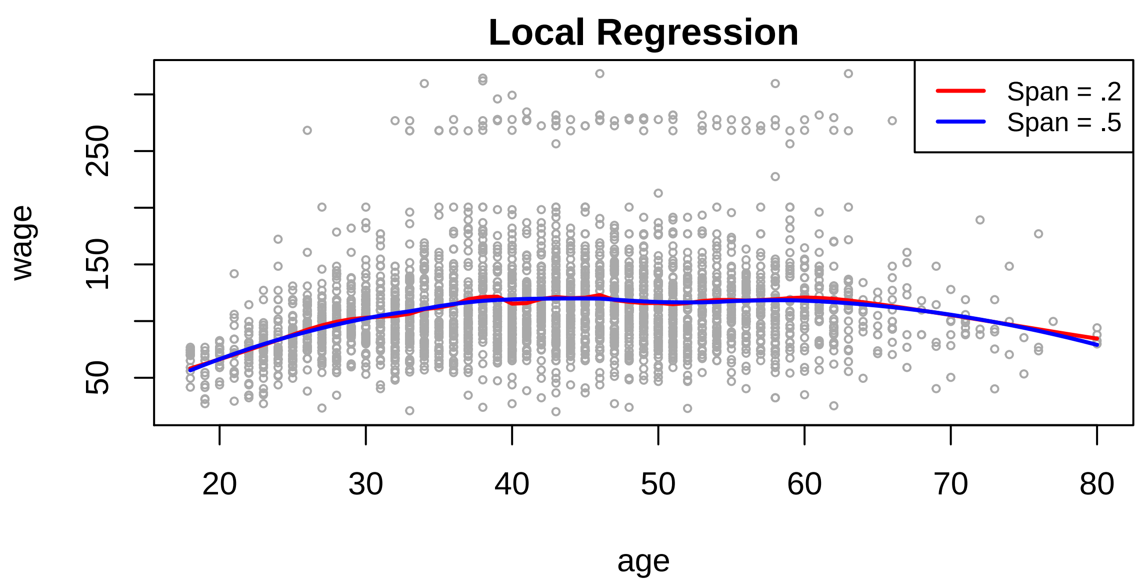
```
par(mar=c(4,4,1.5,.5))
plot(age, wage, xlim = agelims, cex = .5, col = 'darkgrey')
title('Smoothing Spline')
fit <- smooth.spline(age, wage, df = 16)                    # specify df
fit2 <- smooth.spline(age, wage, cv = TRUE)                 # specify Cross-validation
lines(fit, col = 'red', lwd = 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)
```



Local Regression

To fit a *local regression*: we use `loess()`, to specify the `percentage` of observations for each neighborhood, like 20% `span = .2`:

```
par(mar=c(4,4,1.5,.5))
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 = .2', 'Span = .5'), col = c('red', 'blue'), lty=1, lwd=2, cex=.8)
```



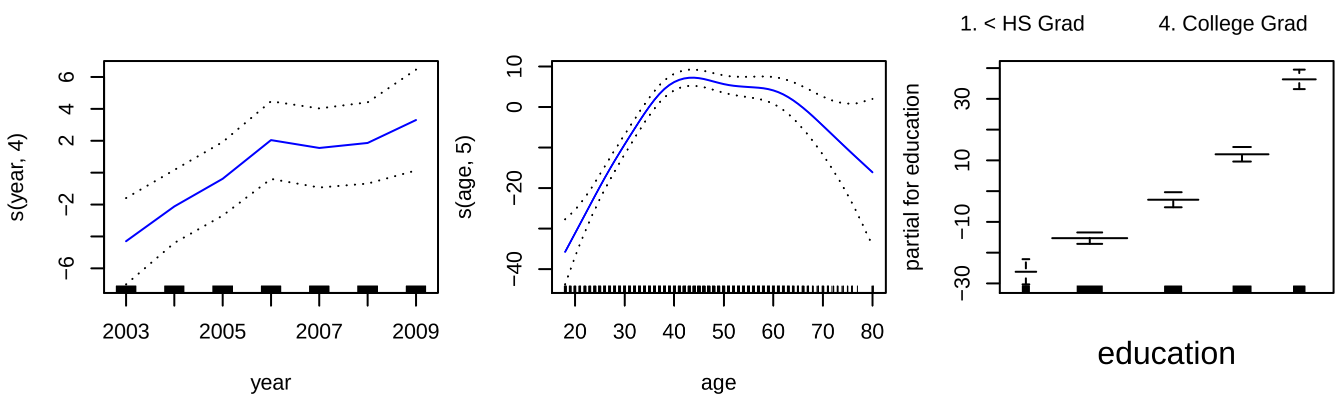
7.8.3. Generalized Additive Models

Goal: fitting a GAM: wage $= f(\text{education}, ns(\text{year}, d=4), ns(\text{age}, d=5)$:

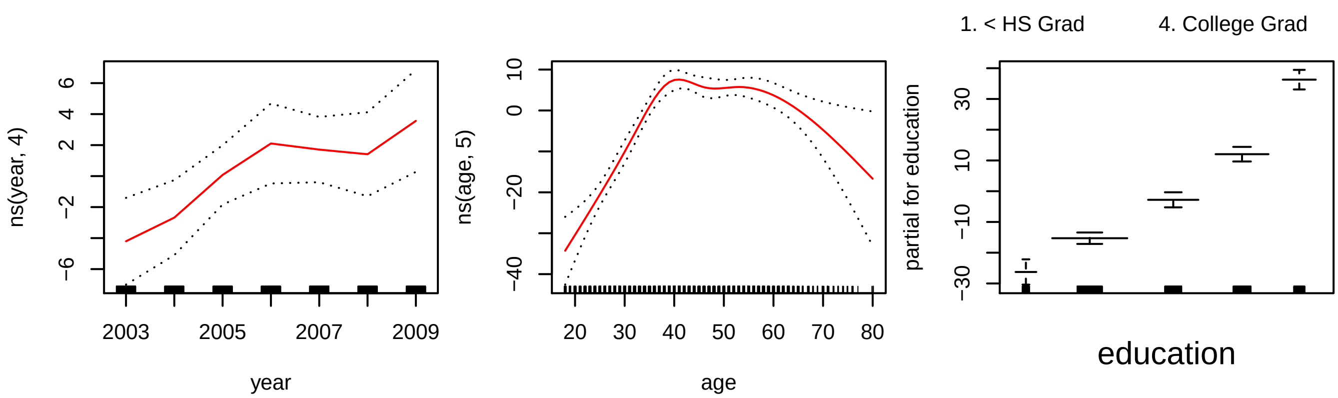
```
gam1 <- lm(wage ~ ns(year,4) + ns(age,5) + education, data = Wage)
```

To fit more GAM using more general basis functions, we use `gam` package, the function `gam()`. To specify a smoothing spline, use `s()`:

```
par(mfrow=c(1,3), mar=c(4,4,2,.5))
library(gam)
gam.m3 <- gam(wage ~ s(year, 4) + s(age, 5) + education, data = Wage)
plot(gam.m3, se=TRUE, col = 'blue')
```




```
par(mfrow=c(1,3), mar=c(4,4,2,.5))
plot.Gam(gam1, se=TRUE, col = 'red')
```



Problem: choosing the best model between:

- Model 1 $\mathcal{M}_1$: GAM without `year`
- Model 2 $\mathcal{M}_2$: GAM with a *linear* function of `year`
- Model 3 $\mathcal{M}_3$: GAM with a *spline* function of `year`

```r
gam.m1 <- gam(wage ~ s(age,5) + education, data = Wage)
gam.m2 <- gam(wage ~ year + s(age,5) + education, data = Wage)
anova(gam.m1, gam.m2, gam.m3, test = 'F')
```

```
## Analysis of Deviance Table
##
## Model 1: wage ~ s(age, 5) + education
## Model 2: wage ~ year + s(age, 5) + education
## Model 3: wage ~ s(year, 4) + s(age, 5) + education
##   Resid. Df Resid. Dev Df Deviance       F    Pr(>F)
## 1      2990    3711731
## 2      2989    3693842  1  17889.2 14.4771 0.0001447 ***
## 3      2986    3689770  3   4071.1  1.0982 0.3485661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation: evidence that $\mathcal{M}_2$ is better than $\mathcal{M}_1$ but $mathcalM_2$ and $\mathcal{M}_3$ are not significantly different $\implies$ $\mathcal{M}_2$ is preferred.

```r
summary(gam.m3)
```

```
##
## Call: gam(formula = wage ~ s(year, 4) + s(age, 5) + education, data = Wage)
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -119.43  -19.70   -3.33   14.17  213.48
##
## (Dispersion Parameter for gaussian family taken to be 1235.69)
##
##     Null Deviance: 5222086 on 2999 degrees of freedom
## Residual Deviance: 3689770 on 2986 degrees of freedom
## AIC: 29887.75
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##               Df  Sum Sq Mean Sq F value                Pr(>F)
## s(year, 4)     1   27162   27162  21.981           0.000002877 ***
## s(age, 5)      1  195338  195338 158.081 < 0.00000000000000022 ***
## education      4 1069726  267432 216.423 < 0.00000000000000022 ***
## Residuals   2986 3689770    1236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df Npar F               Pr(F)
## (Intercept)
## s(year, 4)        3  1.086              0.3537
## s(age, 5)         4 32.380 <0.0000000000000002 ***
## education
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpretation: at the `ANOVA for Nonparametric Effects` from summary table above:

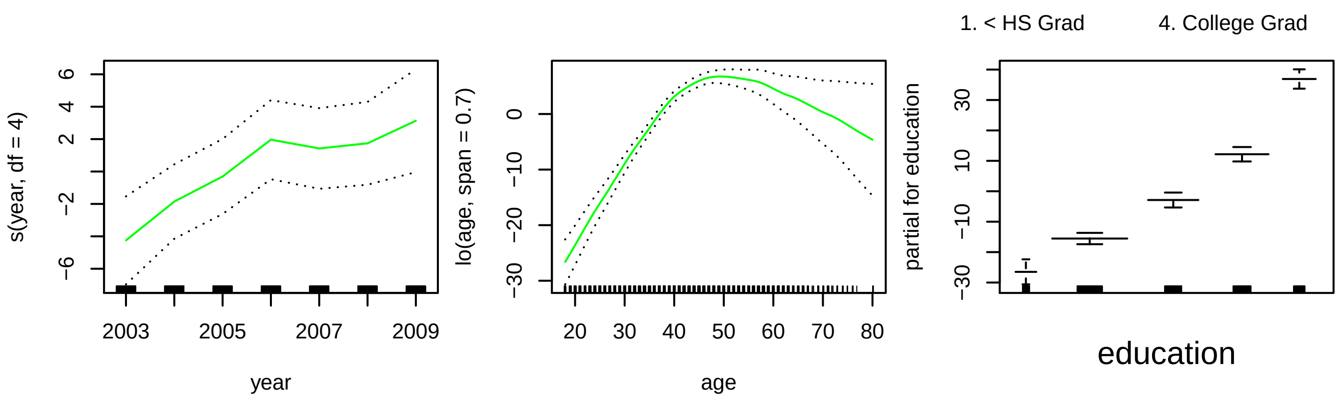
- p-value for `age` and `year` is of H_0: linear relationship vs. H_1: non-linear
- $p = .3537$ indicates linear function is enough for `year`
- $p \approx 0$ indicates a non-linear function is preferred for `age`

```
preds <- predict(gam.m2, newdata = Wage)
```

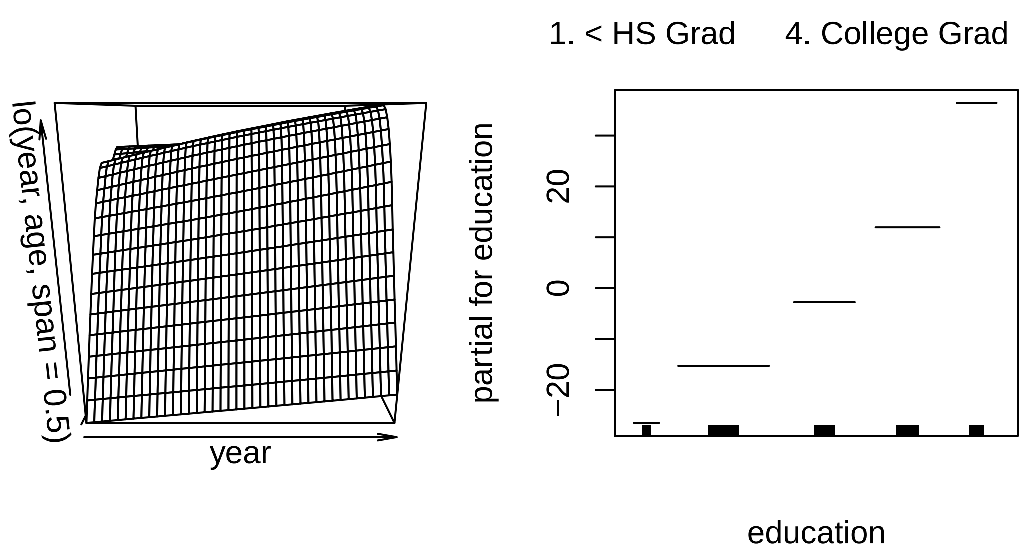
Local Regression

Alternatively, to fit a *local regression*, `lo()`:

```
par(mfrow=c(1,3), mar=c(4,4,2,.5))
gam.lo <- gam(wage ~ s(year, df=4) + lo(age, span=.7) + education, data = Wage)
plot.Gam(gam.lo, se=TRUE, col = 'green')
```

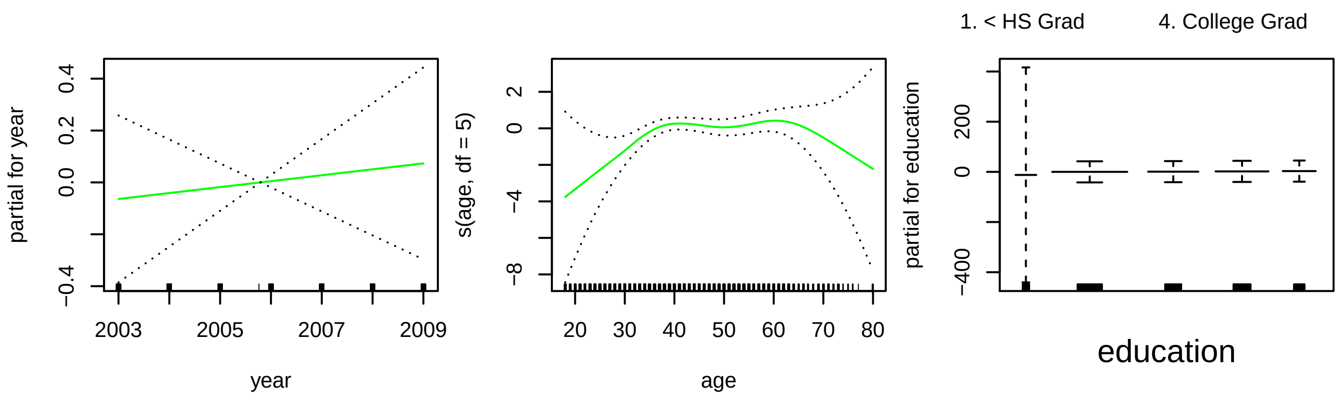



```
par(mfrow=c(1,2), mar=c(4,4,2,.5))
gam.lo.i <- gam(wage ~ lo(year, age, span = .5) + education, data = Wage)
library(akima)
plot(gam.lo.i)
```



To fit a *logistic regression GAM*:

```
par(mfrow=c(1,3), mar=c(4,4,2,.5))
gam.lr <- gam(I(wage>250) ~ year + s(age,df=5) + education, family = binomial, data = Wage)
plot(gam.lr, se=T, col = 'green')
```




```
table(education, I(wage>250))
```

```
##
## education           FALSE TRUE
##   1. < HS Grad        268    0
##   2. HS Grad          966    5
##   3. Some College     643    7
##   4. College Grad     663   22
##   5. Advanced Degree  381   45
```

```
par(mfrow=c(1,3), mar=c(4,4,2,.5))
gam.lr.s <- gam(I(wage>250) ~ year + s(age,df=5) + education, family = binomial,
                data = Wage, subset = (education != '1.  < HS Grad'))
plot(gam.lr.s, se=T, col = 'green')
```