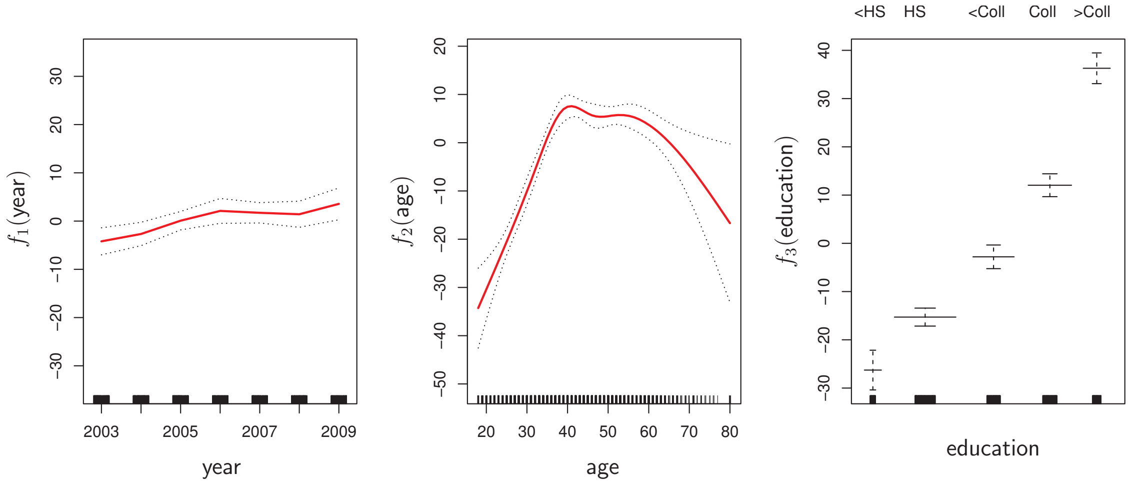
- We fit the first two functions using natural splines. We fit the third function using a separate constant for each level, via the usual dummy variable approach.



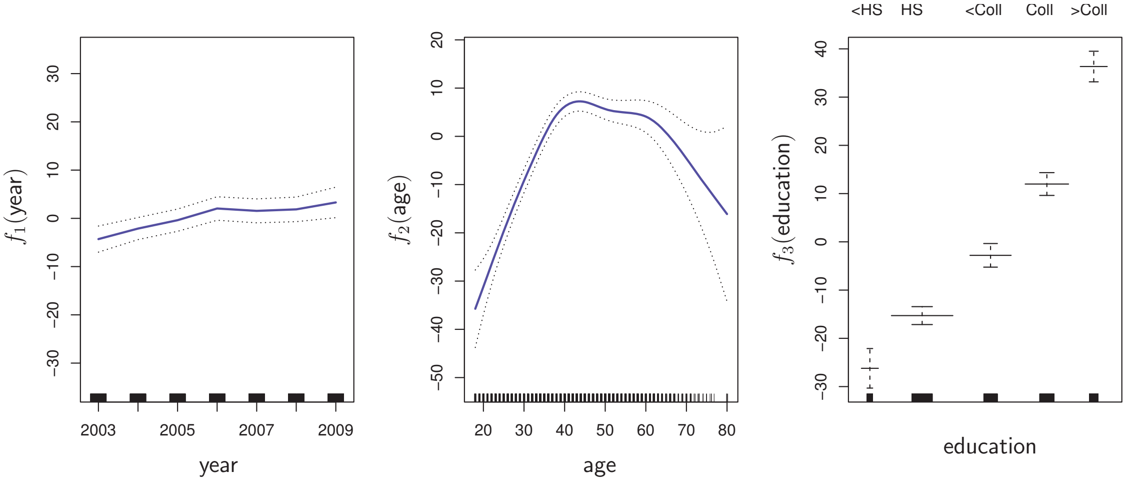
For the Wage data, plots of the relationship between each feature and the response, wage. Each plot displays the fitted function and pointwise standard errors. The first two functions are natural splines in year and age, with four and five degrees of freedom, respectively. The third function is a step function, fit to the qualitative variable education.

Interpretation:

The left-hand panel indicates that holding age and education fixed, wage tends to increase slightly with year. This may be due to inflation.

The center panel indicates that holding education and year fixed, wage tends to be highest for intermediate values of age, and lowest for the very young and very old.

The right-hand panel indicates that holding year and age fixed, wage tends to increase with education: the more educated a person is, the higher their salary, on average.



Details are as in previous figure, but now f_1 and f_2 are smoothing splines with four and five degrees of freedom, respectively.

Note: We do not have to use splines as the building blocks for GAMs: we can just as well use local regression, polynomial regression, or any combination of the approaches seen earlier in this chapter in order to create a GAM.

Pros and Cons of GAMs:

- GAMs allow us to fit a non-linear f_j to each X_j, so that we can automatically model non-linear relationships that standard linear regression will miss.
- The non-linear fits can potentially make more accurate predictions for the response Y.
- Because the model is additive, we can examine the effect of each X_j on Y individually while holding all of the other variables fixed.

- The smoothness of the function f_j for the variable X_j can be summarized via degrees of freedom.
- The main limitation of GAMs is that the model is restricted to be additive. With many variables, important interactions can be missed.
- However, as with linear regression, we can manually add interaction terms to the GAM model by including additional predictors of the form $X_j \times X_k$.

GAMs for Classification Problems

- GAMs can also be used in situations where Y is qualitative. For simplicity, here we will assume Y takes on values zero or one, and let $p(X) = Pr(Y = 1|X)$ be the conditional probability (given the predictors) that the response equals one.
- Recall the logistic regression model:

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p$$

- A natural way to extend the model above to allow for non-linear relationships is to use the model

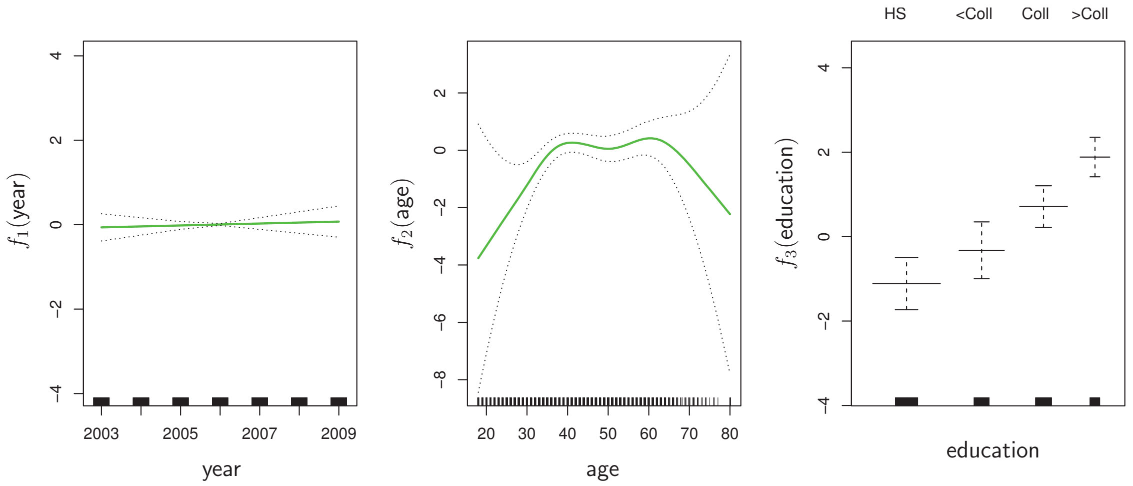
$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + f_1(X_1) + f_2(X_2) + \ldots + f_p(X_p)$$

This is a logistic regression GAM.

- For example, we fit a GAM to the Wage data in order to predict the probability that an individual's income exceeds $250,000 per year. The GAM that we fit takes the form

$$\log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 \times \text{year} + f_2(\text{age}) + f_3(\text{education})$$

$p(X) = Pr(wage > 250 | year, age, education)$



For the Wage data, the logistic regression GAM given in (7.19) is fit to the binary response I(wage $>$ 250). Each plot displays the fitted function and pointwise standard errors. The first function is linear in year, the second function a smoothing spline with five degrees of freedom in age, and the third a step function for education.

Performing GAMs in R

We now fit a GAM to predict wage using natural spline functions of year and age, treating education as a qualitative predictor, as in

$$\text{wage} = \beta_0 + f_1(\text{year}) + f_2(\text{age}) + f_3(\text{education}) + \epsilon$$

Since this is just a big linear regression model using an appropriate choice of basis functions, we can simply do this using the lm() function.

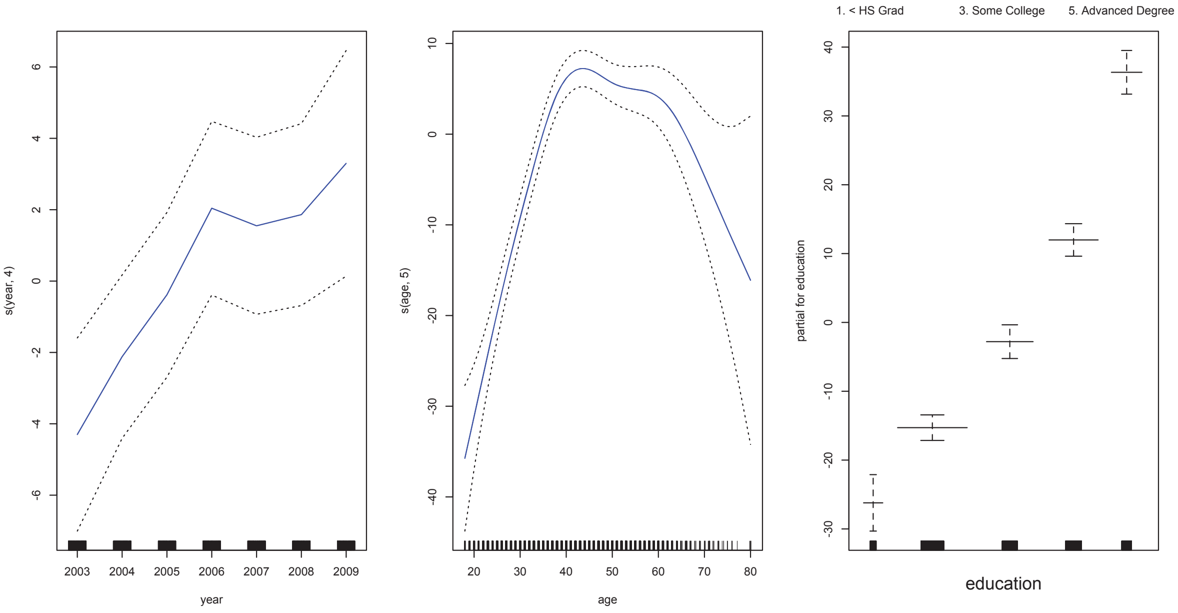
```
gam1 <- lm(wage ~ ns(year, 4) + ns(age, 5) + education, data = Wage)
```

In order to fit more general sorts of GAMs, using smoothing splines or other components that cannot be expressed in terms of basis functions and then fit using least squares regression, we will need to use the gam() function which is part of the gam library in R.

The s() function, which is also part of the gam library, is used to indicate that we would like to use a smoothing spline.

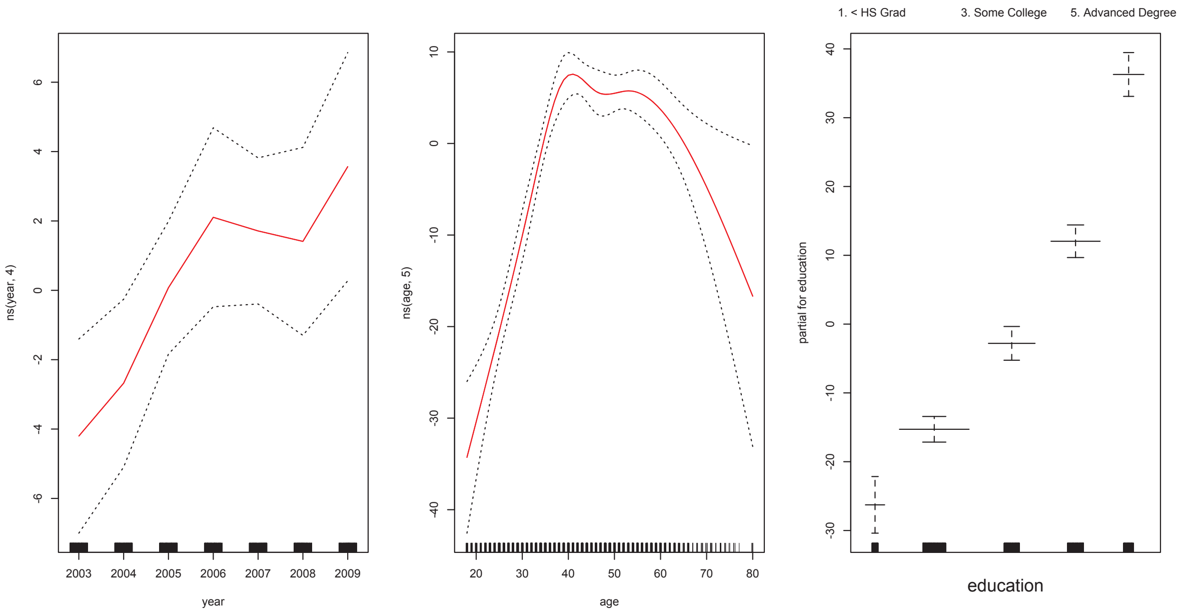
```
> library(gam)
> gam_m3 <- gam(wage ~ s(year, 4) + s(age, 5) + education, data = Wage)
> par(mfrow = c(1, 3))
> plot(gam_m3, se = TRUE, col = "blue")
```

The generic plot() function recognizes that gam_m3 is an object of class Gam, and invokes the appropriate plot.Gam() method.



Conveniently, even though gam1 is not of class Gam but rather of class lm, we can still use plot.Gam() on it.

```
> plot.Gam(gam1, se = TRUE, col = "red")
```



In these plots, the function of year looks rather linear. We can perform a series of ANOVA tests in order to determine which of these three models is best: a GAM that excludes year ($\mathcal{M}_1$), a GAM that uses a linear function of year ($\mathcal{M}_2$), or a GAM that uses a spline function of year ($\mathcal{M}_3$).

```
> 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
```

we find that there is compelling evidence that a GAM with a linear function of year is better than a GAM that does not include year at all (P-value = 0.0001447). However, there is no evidence that a non-Linear function of year is needed (P-value=0.349). In other words, based on the results of this ANOVA, M_2 is preferred.

The summary() function produces a summary of the gam fit.

```
> 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 2.877e-06 ***
s(age, 5)     1  195338  195338 158.081 < 2.2e-16 ***
education     4 1069726  267432 216.423 < 2.2e-16 ***
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 <2e-16 ***
education                            
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The “Anova for Parametric Effects” p-values clearly demonstrate that year, age, and education are all highly statistically significant, even when only assuming a linear relationship.
Alternatively, the “Anova for Nonparametric Effects” p-values for year and age correspond to a null hypothesis of a linear relationship versus the alternative of a non-linear relationship. The large p-value for year reinforces our conclusion from the ANOVA test that a linear function is adequate for this term. However, there is very clear evidence that a non-linear term is required for age.

We can make predictions using the predict() method for the class Gam. Here we make predictions on the training set.

```
> preds <- predict(gam_m2, newdata = Wage)
```

We can also use local regression fits as building blocks in a GAM, using the lo() function.

```
> gam_lo <- gam(wage ~ s(year, df = 4) + lo(age, span = 0.7) + education,
   data = Wage)
> plot.Gam(gam_lo, se = TRUE, col = "green")
```