7.8.3 GAMs

Team 13

2/17/2021

```
rm(list=ls())
library(ISLR)
library(splines)
attach(Wage)
```

GAM – Generalized Additive Models

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

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

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.

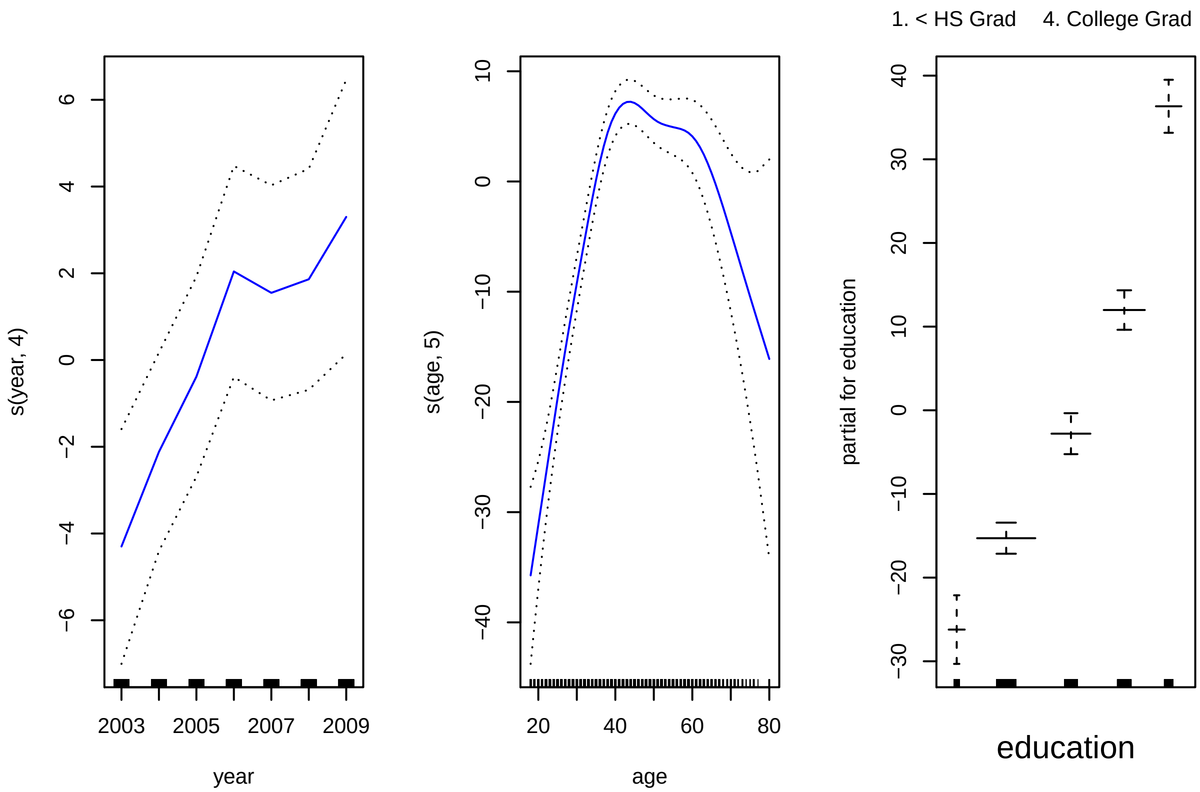
We now fit the model using smoothing splines rather than natural splines. In order to fit more general sorts of GAMS, using smoothing splines or other components that cannot be expressed in terms of basis functions and then fit using least squares regression, we will need to use the gam library in R.

Year should have 4 degrees of freedom, age should have 5 degrees of freedom, education is qualitative – leave it as is.

```
library(gam)
```

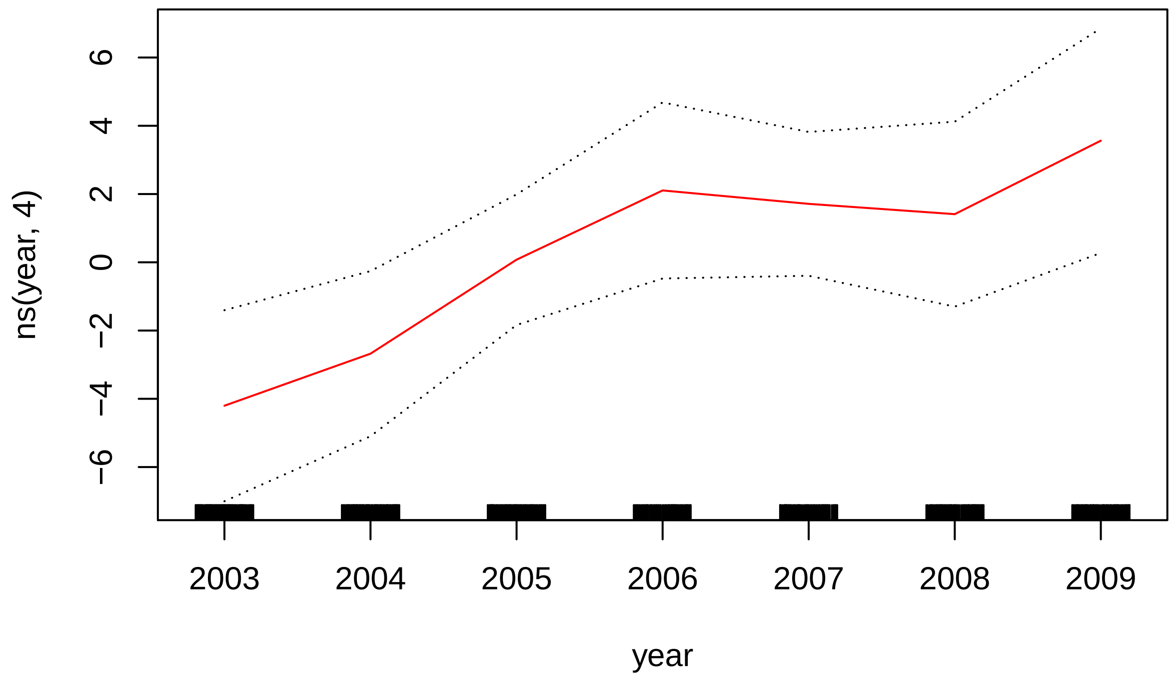
```
## Warning: package 'gam' was built under R version 4.0.3

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.0.3

## Loaded gam 1.20
```

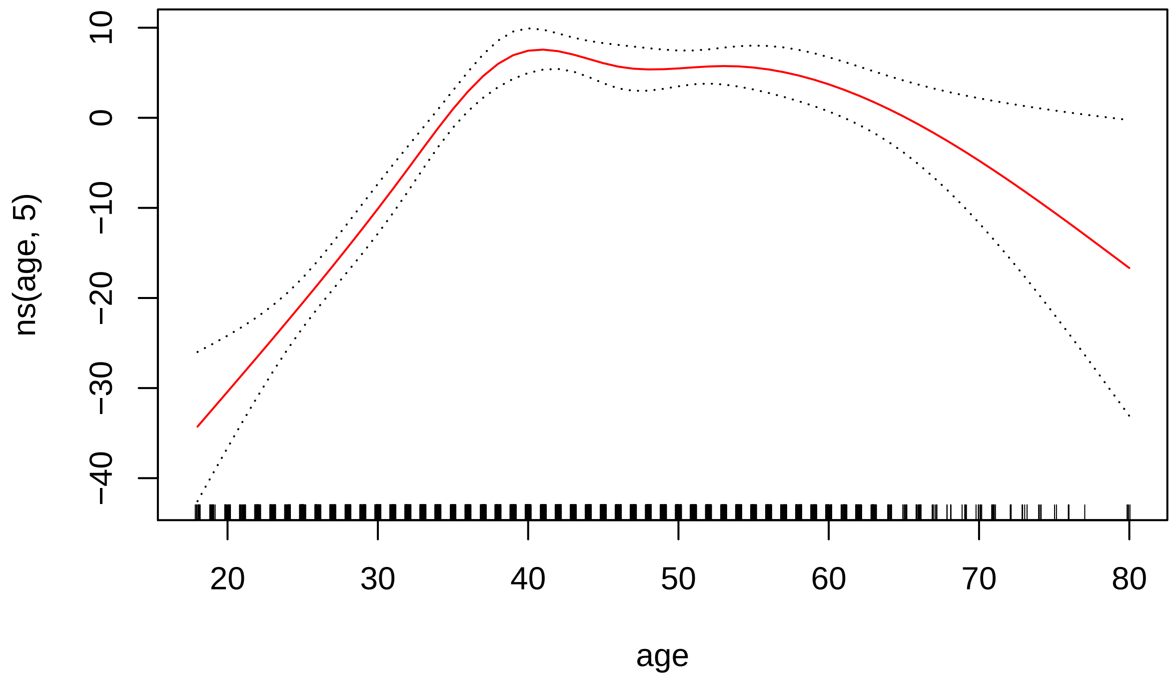
```
gam.m3=gam(wage~s(year,4)+s(age,5)+education, data=Wage)
par(mfrow=c(1,3))
plot(gam.m3, se=TRUE, col="blue")
```



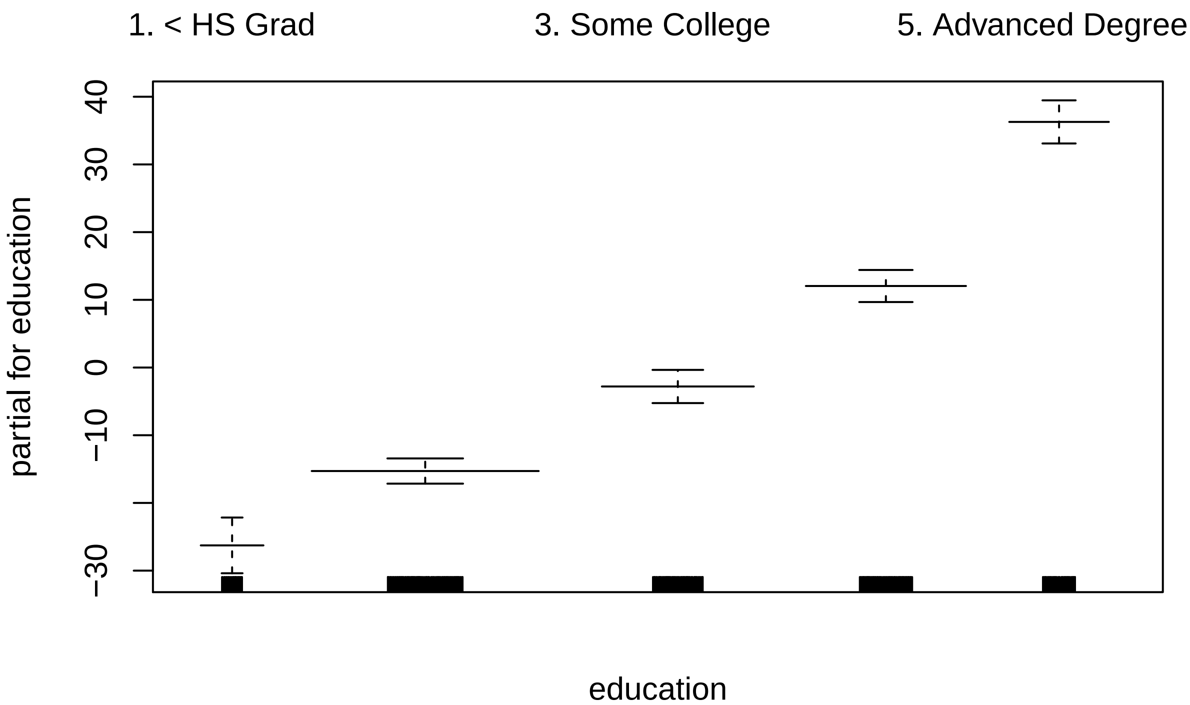
The general plot() function recognizes gam.m3 as an object of class gam, and invokes the appropriate plot.gam()method. Even gam1 is not of class gam but rather of class lm, we can still use plot.gam() on it. Plot.Gam reproduces figure 7.11 on page 284, instead of the general plot function.

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


ns(year, 4)
6
4
2
0
−2
−4
−6
2003
2004
2005
2006
2007
2008
2009
year



ns(age, 5)
10
0
−10
−20
−30
−40
20
30
40
50
60
70
80
age



Note that the proper syntax on the current version R is plot.Gam() instead of lower case plot.gam.

In these plots, the function of year looks rather linear. We can perform a series of ANOVA tests in order to determin which of these three models is best: a GAM that excludes year, a GAM that usese a linear function of year, or a GAM that uses a spline function of year.

```
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
```

From the ANOVA tests above 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=0.00014). 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, m2 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: NA
##
## 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
```

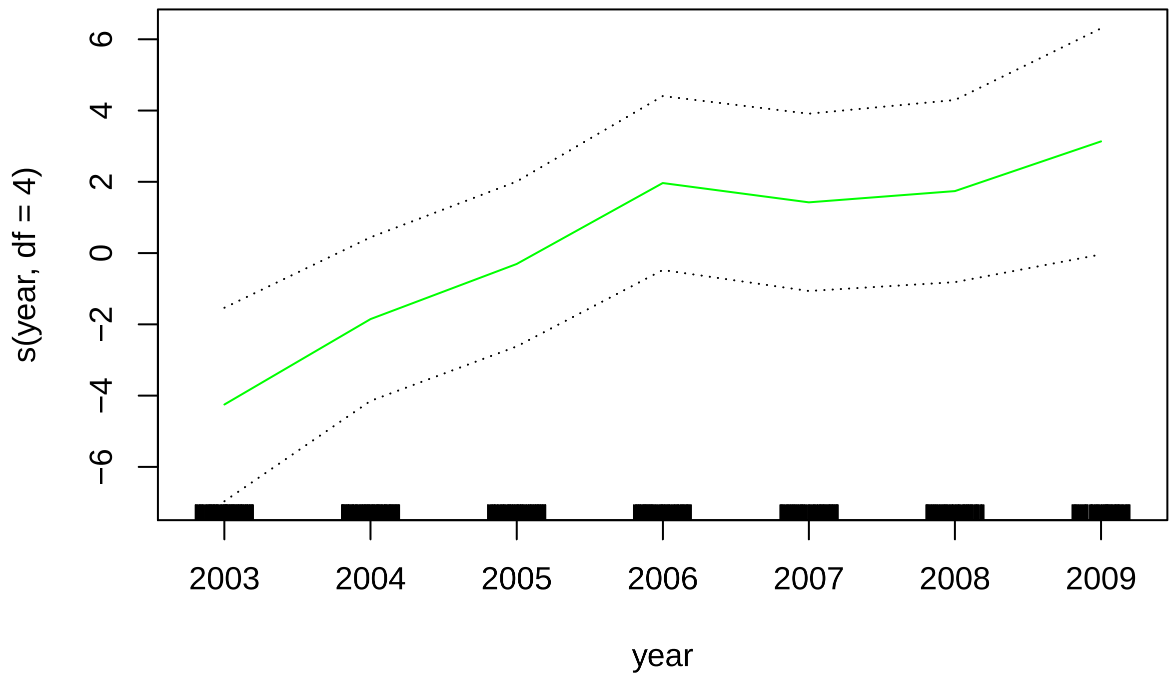
The p-value of year and age correspond to a null hypothesis of a linear relationship versus the alternative of a non-linear relationship. The large p-value for year reinforces our conclusion from the ANOVA test that a linear function is adequate for this term. However, there is ver clear evidence that a non-linear term is required for age.

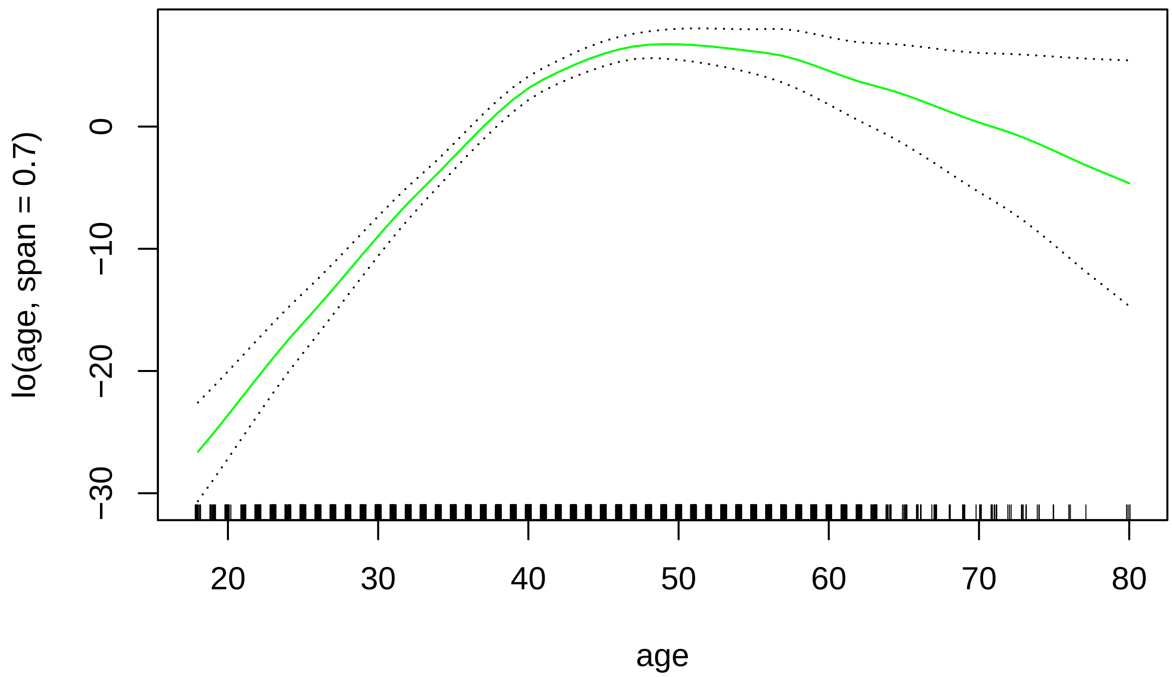
We can make predictions from gam objects, just like from lm objects, using the predict() method for the class gam. Here we make predictions on the training set for the best model – m2.

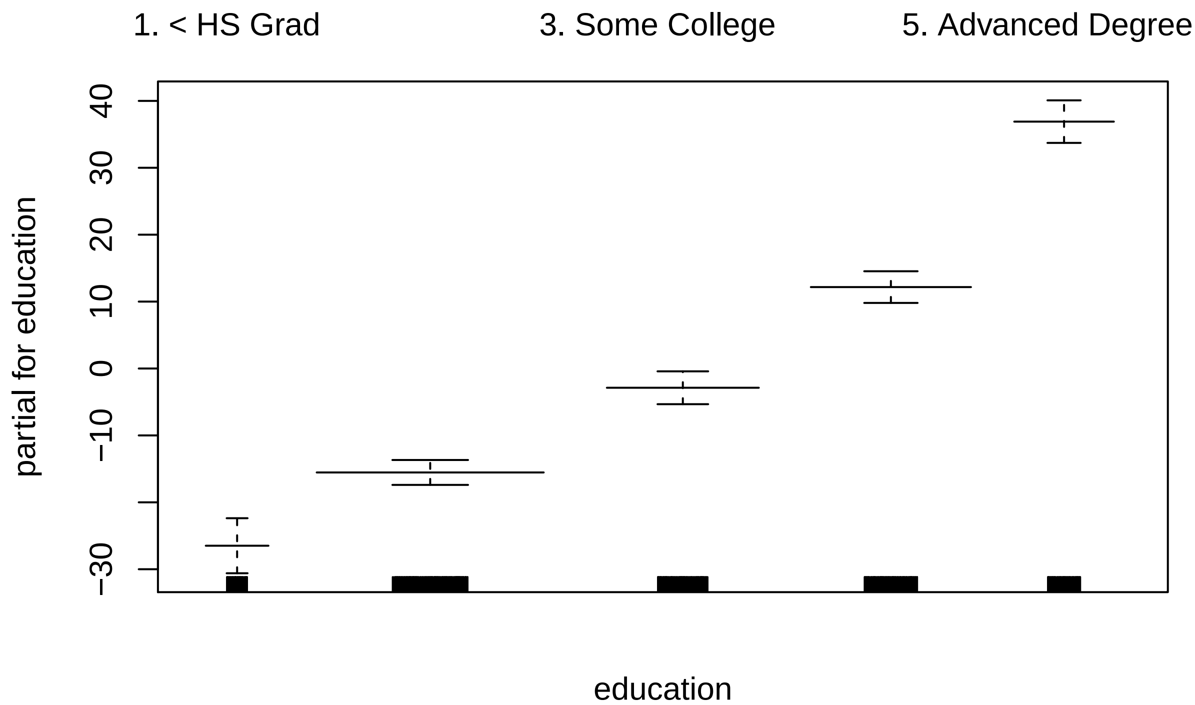
```r
preds=predict(gam.m2, newdata=Wage)
```

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

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


s(year, df = 4)
6
4
2
0
−2
−4
−6
2003
2004
2005
2006
2007
2008
2009
year


lo(age, span = 0.7)
0
−10
−20
−30
20
30
40
50
60
70
80
age



We can also use the lo() function to create interactions before calling the gam() function.

```
gam.lo.i=gam(wage~lo(year,age,span=0.5)+education, data=Wage)
```

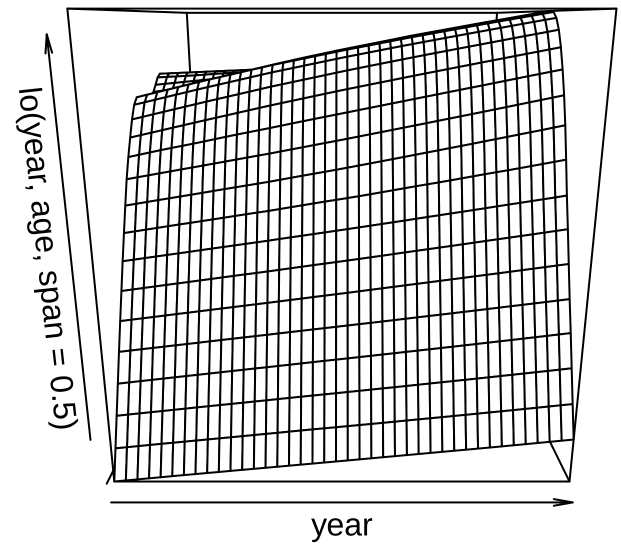
```
## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, : liv
## too small. (Discovered by lowesd)

## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, : lv
## too small. (Discovered by lowesd)

## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, : liv
## too small. (Discovered by lowesd)

## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, : lv
## too small. (Discovered by lowesd)
```

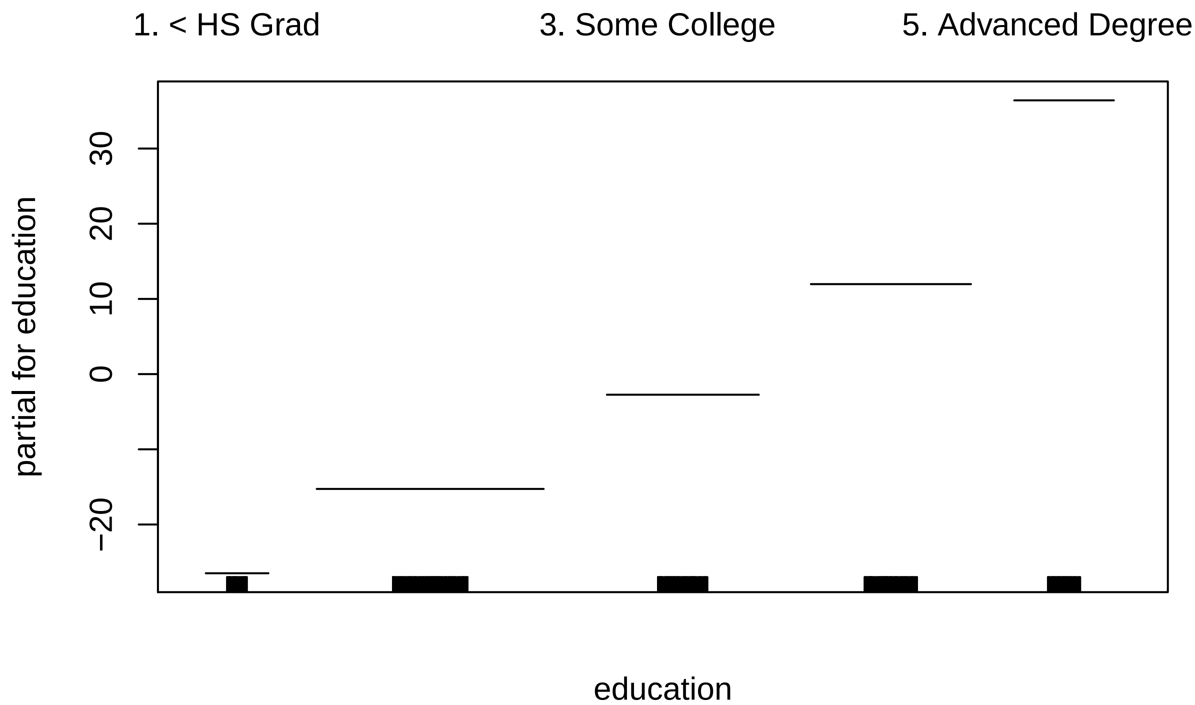
It fits a two-term model, in which the first term is an interaction between year and age, fit by a local regression surface. We can plot the resulting two-dimensional surface if we first install the akima package.

```
library(akima)
```

```
## Warning: package 'akima' was built under R version 4.0.3
```

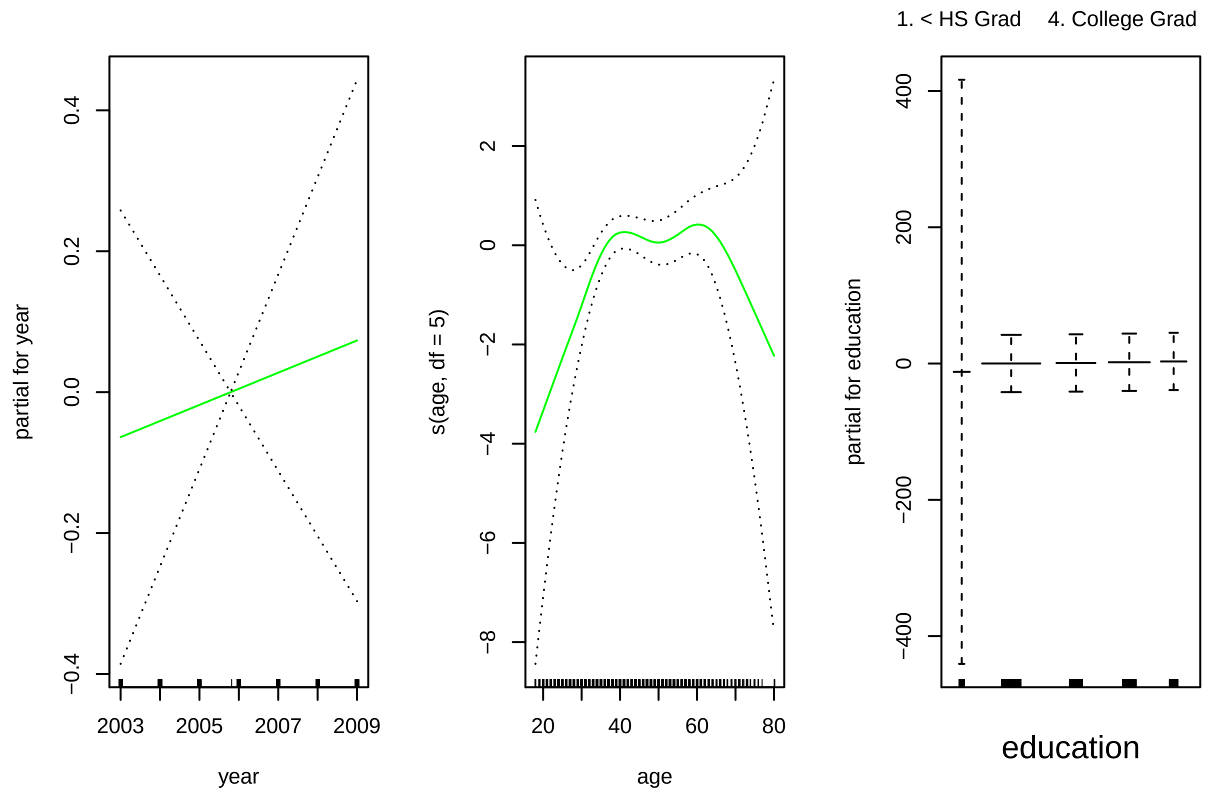
```r
plot(gam.lo.i)
```





In order to fit a logistic regression GAM, we once again use the I() function in constructing the binary response variable, and set family=binomial.

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



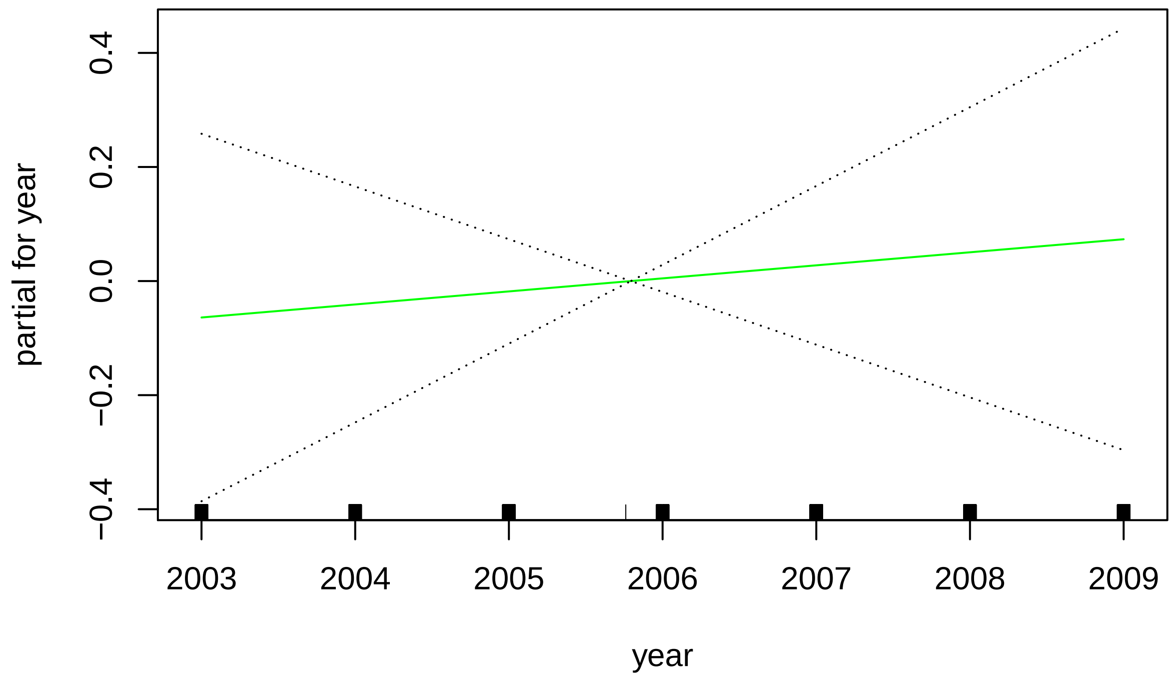
It is easy to see that there are no high earners in the <HS category

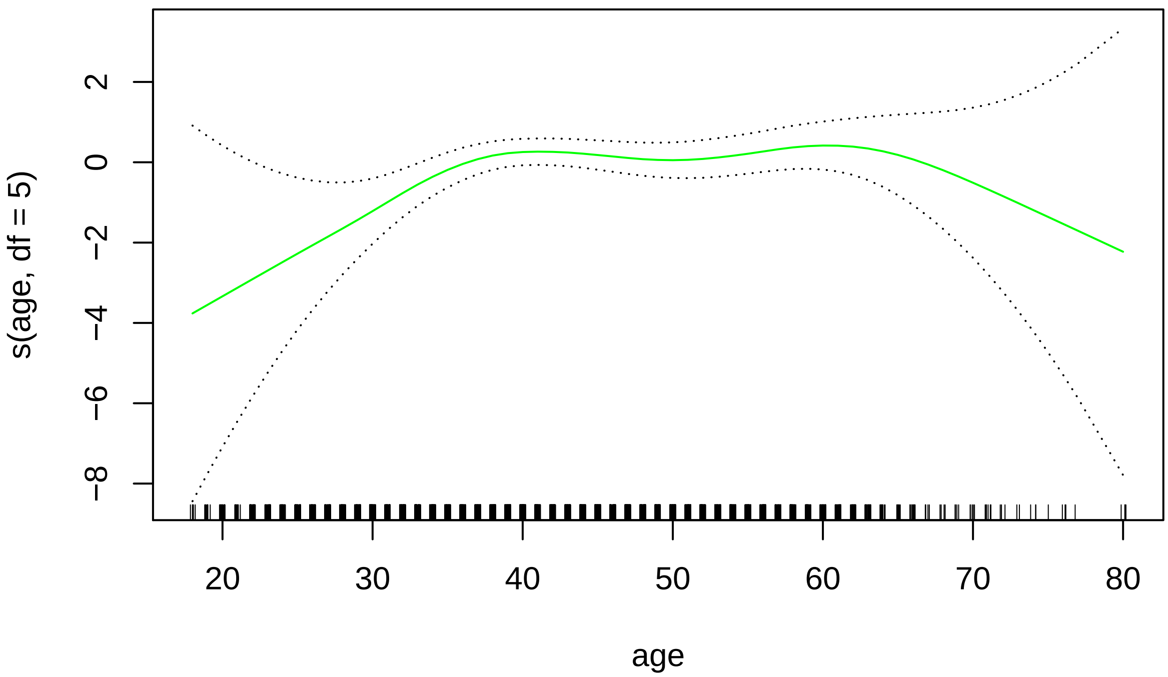
```r
table(education, I(wage>250))
```

```
##
## 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
```

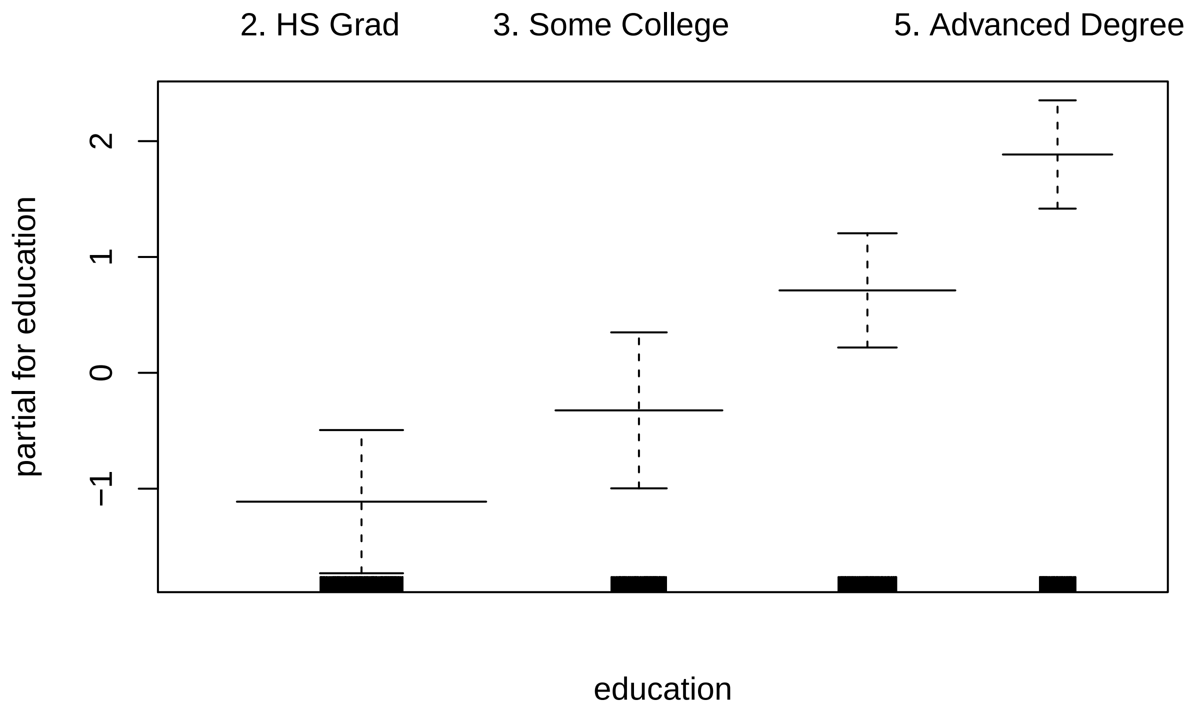
Hence we fit a logistic regression GAM using all but this category. This provides more sensible results.

```r
gam.lr.s=gam(I(wage>250)~year+s(age,df=5)+education, family=binomial, data=Wage, subset=(education!="1.
plot(gam.lr.s,se=T,col="green")
```


partial for year
0.4
0.2
0.0
−0.2
−0.4
2003
2004
2005
2006
2007
2008
2009
year



2
0
−2
−4
−6
−8
s(age, df = 5)
20
30
40
50
60
70
80
age



2. HS Grad
3. Some College
5. Advanced Degree
2
1
0
−1
partial for education
education