R Lab 8

Jared Andreatta

2025-04-28

7.8.2 Splines

Use the splines library.

```
library(splines)
library(ISLR2)
attach(Wage)
agelims <- range(age)
age.grid <- seq(from = agelims [1], to = agelims [2])
# Regression spline fit with 3 knots
# Note the bs() function generates a matrix of basis functions for splines with
# a specified set of knots.
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")

dim(bs(age , knots = c(25, 40, 60)))
```

```
## [1] 3000    6
```

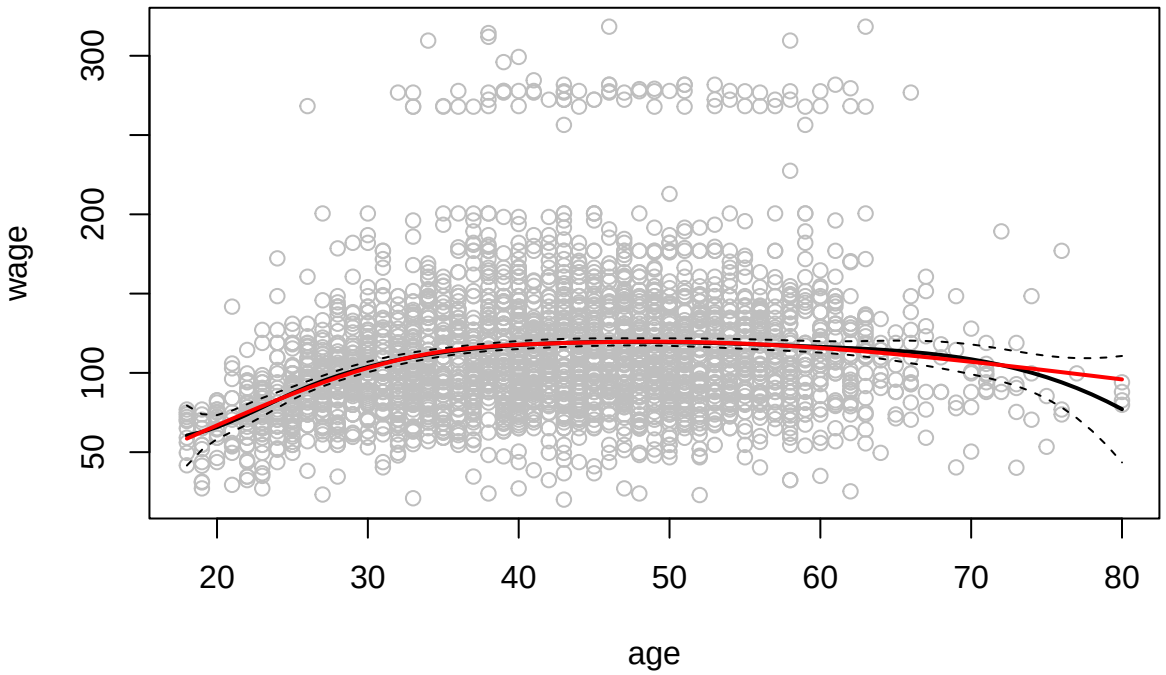
```
dim(bs(age , df = 6))
```

```
## [1] 3000    6
```

```
# R chooses 33.8, 42, and 51 as knots, which correspond to 25, 50, and 75
# quantiles.
attr(bs(age , df = 6), "knots")
```

```
## [1] 33.75 42.00 51.00
```

```
### Fitting with a natural spline with 4 Dfs via ns()
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)
```



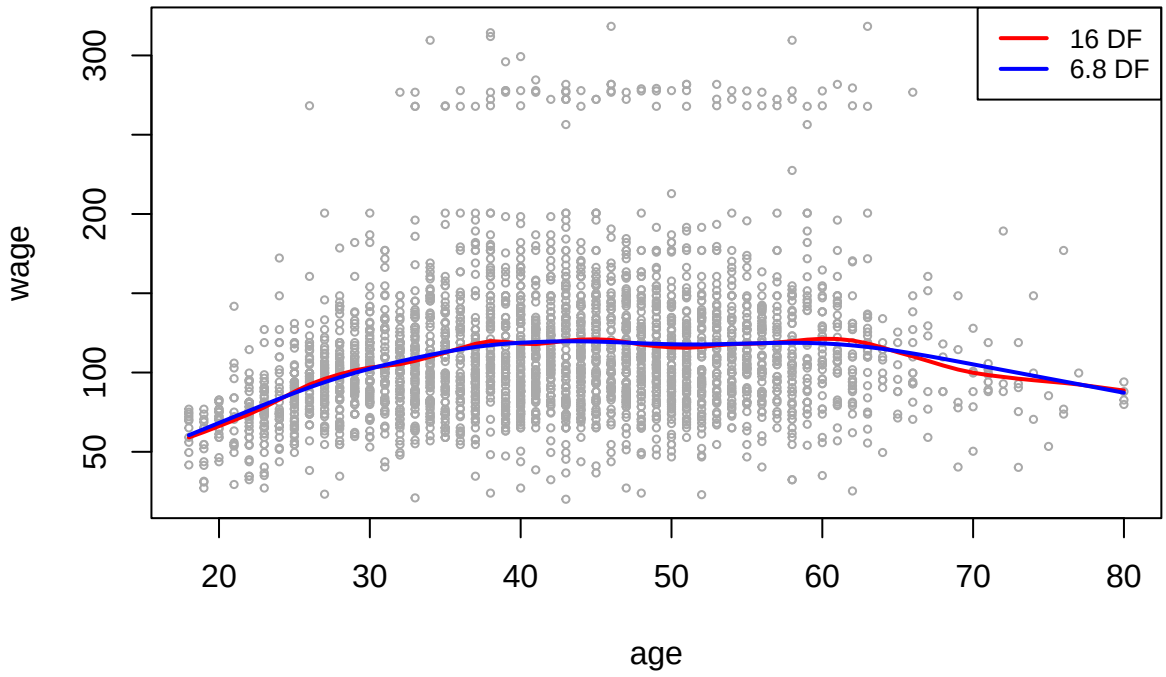

```
### Fitting with smoothing splines via smooth.spline()
plot(age , wage , xlim = agelims , cex = .5, col = "darkgrey")
title("Smoothing Spline")
fit <- smooth.spline(age , wage , df = 16)
fit2 <- smooth.spline(age , wage , cv = TRUE)
```

```
## Warning in smooth.spline(age, wage, cv = TRUE): cross-validation with
## non-unique 'x' values seems doubtful
```

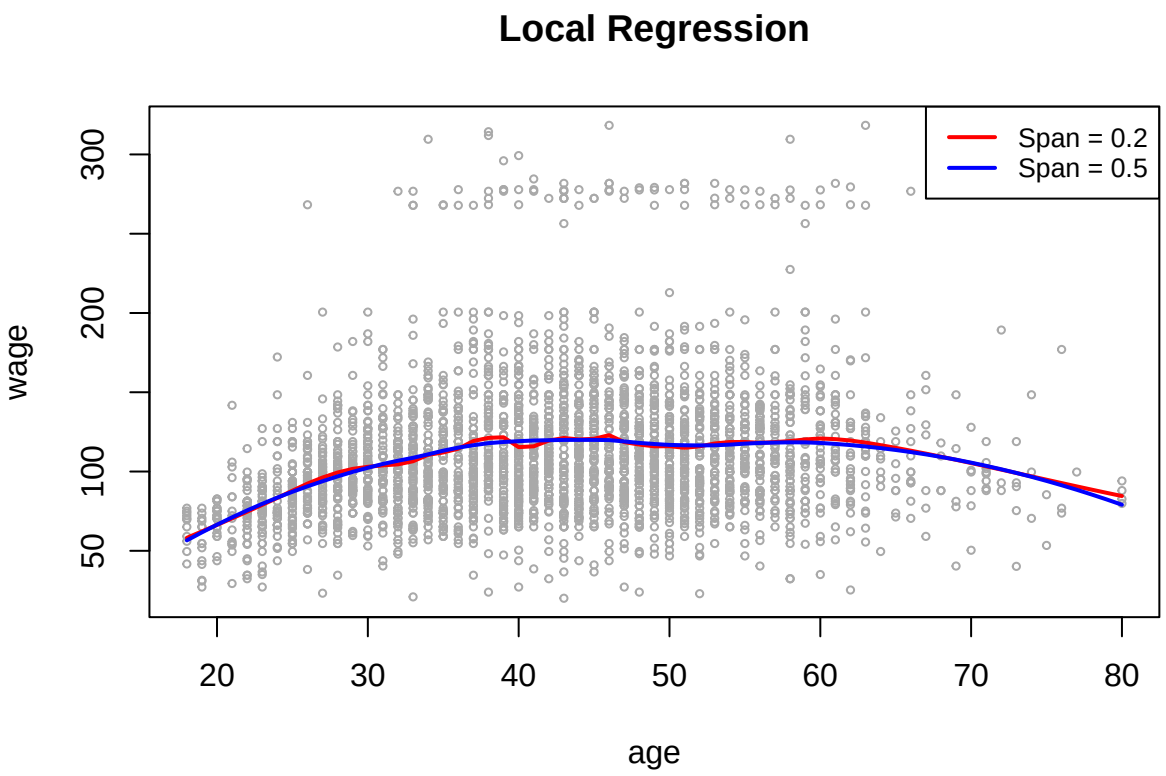
```
fit2$df
```

```
## [1] 6.794596
```

```
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)
```




```
### Fitting with local regressions via LOESS
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)
```



7.8.3 GAMs

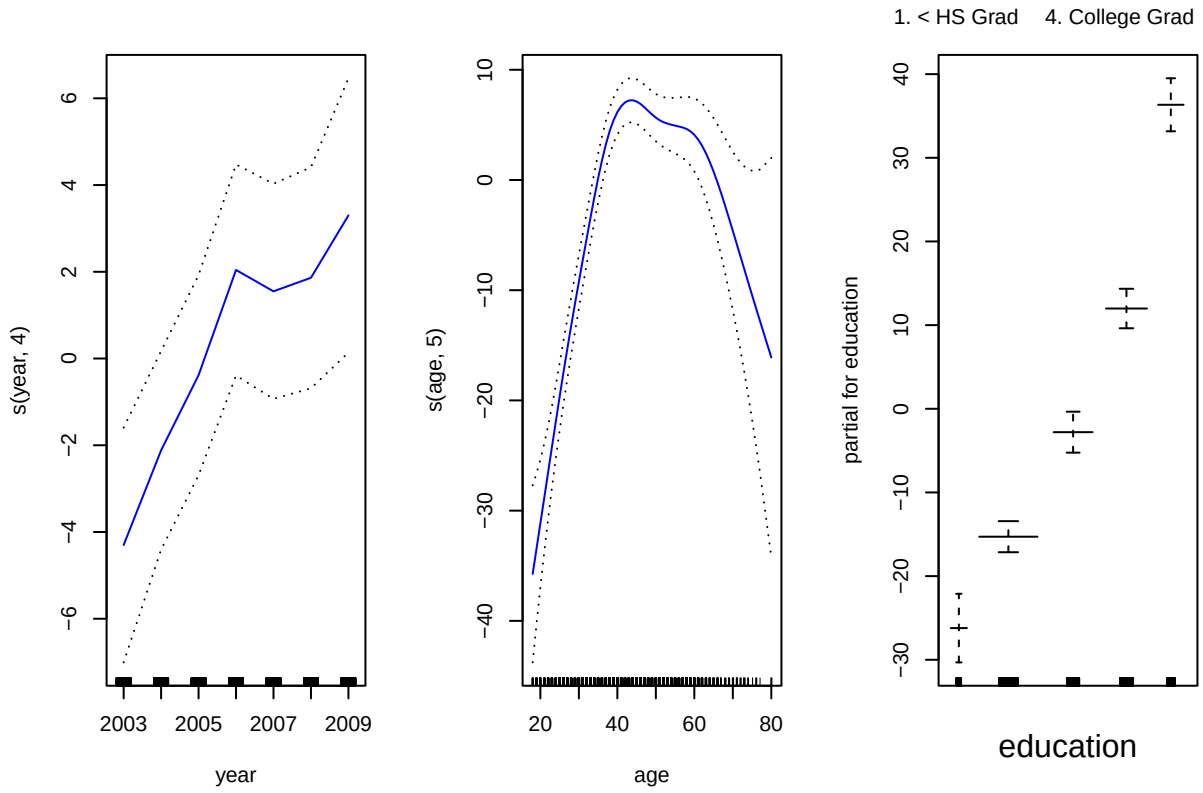
```
# Fitting GAM with natural spline basis functions
gam1 <- lm(wage ~ ns(year , 4) + ns(age , 5) + education ,
data = Wage)

library(gam)
```

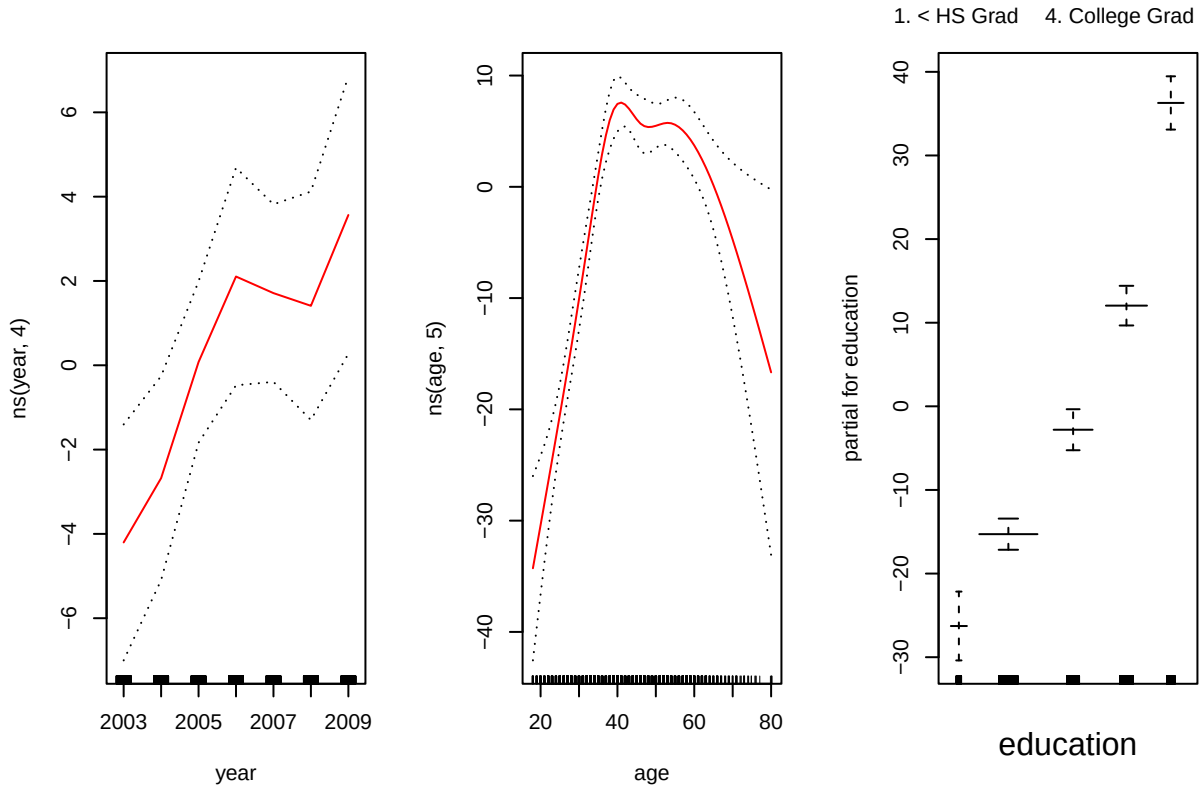
```
## Loading required package: foreach
```

```
## Loaded gam 1.22-5
```

```
# We can use s() to indicate that we want to use smoothing splines
gam.m3 <- gam(wage ~ s(year , 4) + s(age , 5) + education ,
data = Wage)
# Plotting gam object
par(mfrow = c(1, 3))
plot(gam.m3 , se = TRUE , col = "blue")
```




```
# Explicitly calling plot.Gam()
plot.Gam(gam1 , se = TRUE , col = "red")
```



```r
# Performing ANOVA to see which fit is best
# Adding spline function onto year doesn't help
gam.m1 <- gam(wage ~ s(age , 5) + education , data = Wage) # No year for M1
gam.m2 <- gam(wage ~ year + s(age , 5) + education ,
              data = Wage) # Linear function of year for M2
anova(gam.m1 , gam.m2 , gam.m3 , test = "F") # M3 has a spline function for year
```

```
## 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
```

```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
```

```r
# Producing predictions with M2
preds <- predict(gam.m2 , newdata = Wage)
```

```r
# Using lo() for local regressions
gam.lo <- gam( wage ~ s(year , df = 4) + lo(age , span = 0.7) + education ,
               data = Wage)
plot(gam.lo , se = TRUE , col = "green")

# Interaction term inside lo()
gam.lo.i <- gam(wage ~ lo(year , age , span = 0.5) + education ,
data = Wage)
```

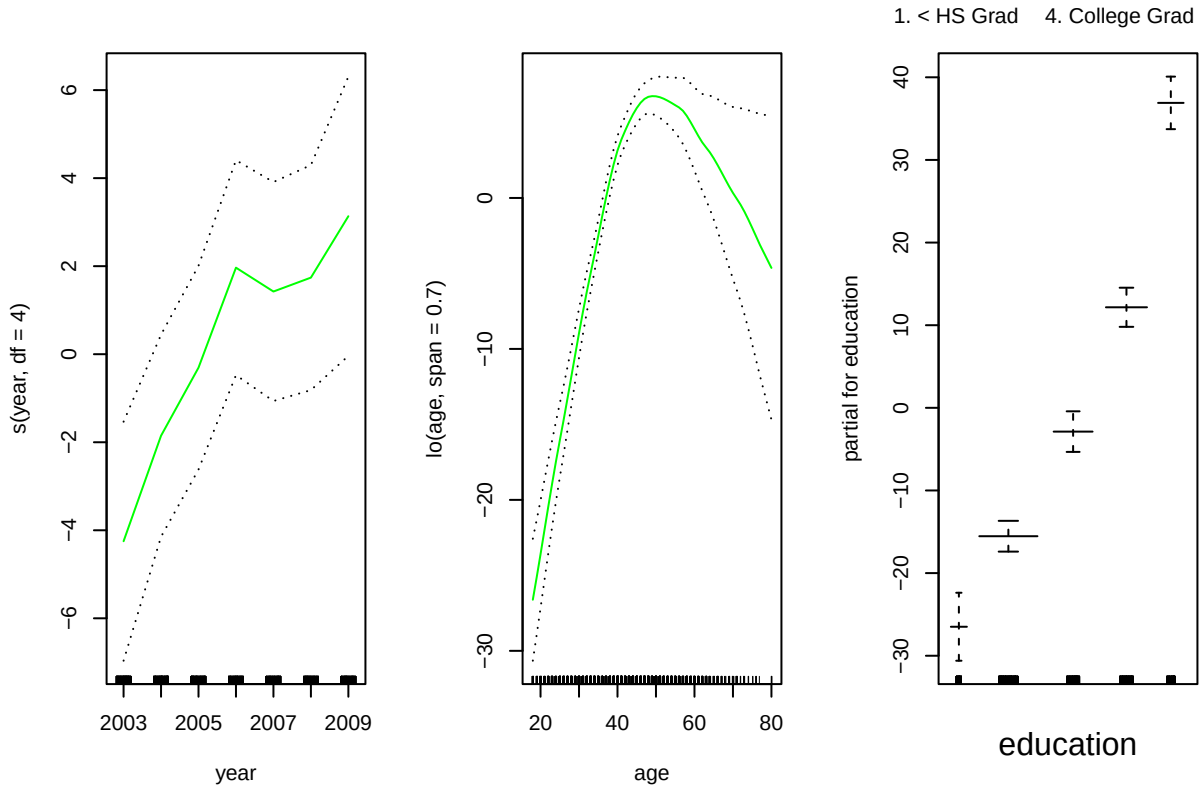
```
## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, :
## liv too small.  (Discovered by lowesd)
```

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

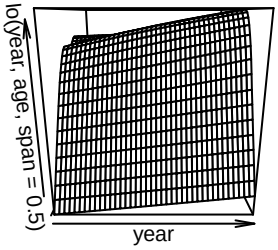
```
## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, :
## liv too small.  (Discovered by lowesd)
```

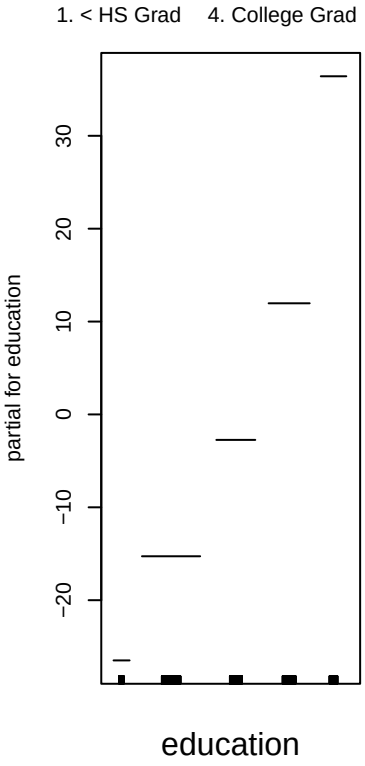
```
## Warning in lo.wam(x, z, wz, fit$smooth, which, fit$smooth.frame, bf.maxit, : lv
## too small.  (Discovered by lowesd)
```

```r
# akima library to plot 2D surface
library(akima)
```




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


lo(year, age, span = 0.5)
year


1. < HS Grad
4. College Grad
partial for education
30
20
10
0
−10
−20
education