Assignment 4

Getrude Gichuhi

2022-06-10

```
#install.packages('npreg')
#install.packages('gam')
#install.packages('ISLR')
```

```
summary(Wage)
```

```
##       year           age                     maritl           race
##  Min.   :2003   Min.   :18.00   1. Never Married: 648   1. White:2480
##  1st Qu.:2004   1st Qu.:33.75   2. Married      :2074   2. Black: 293
##  Median :2006   Median :42.00   3. Widowed      :  19   3. Asian: 190
##  Mean   :2006   Mean   :42.41   4. Divorced     : 204   4. Other:  37
##  3rd Qu.:2008   3rd Qu.:51.00   5. Separated    :  55
##  Max.   :2009   Max.   :80.00
##
##               education                     region               jobclass
##  1. < HS Grad      :268   2. Middle Atlantic   :3000   1. Industrial :1544
##  2. HS Grad        :971   1. New England       :   0   2. Information:1456
##  3. Some College   :650   3. East North Central:   0
##  4. College Grad   :685   4. West North Central:   0
##  5. Advanced Degree:426   5. South Atlantic    :   0
##                           6. East South Central:   0
##                           (Other)              :   0
##             health      health_ins      logwage           wage
##  1. <=Good     : 858   1. Yes:2083   Min.   :3.000   Min.   : 20.09
##  2. >=Very Good:2142   2. No : 917   1st Qu.:4.447   1st Qu.: 85.38
##                                      Median :4.653   Median :104.92
##                                      Mean   :4.654   Mean   :111.70
##                                      3rd Qu.:4.857   3rd Qu.:128.68
##                                      Max.   :5.763   Max.   :318.34
##
```

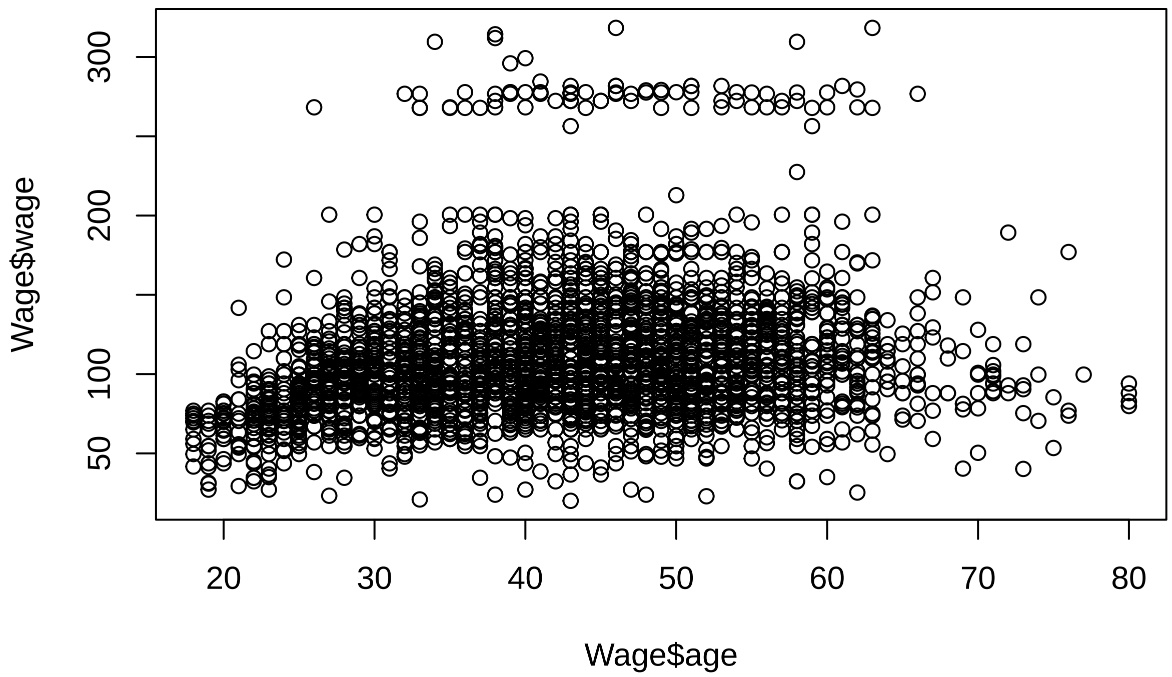
```
dataset = Wage
```

Regression Splines

```
lm_mod = lm(wage ~ year + age, data = Wage)
summary(lm_mod)
```

```
##
## Call:
## lm(formula = wage ~ year + age, data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -96.766 -25.081  -6.108  16.838 209.053
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2318.5309   739.1385  -3.137  0.00172 **
## year            1.1968     0.3685   3.247  0.00118 **
## age             0.6992     0.0647  10.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.86 on 2997 degrees of freedom
## Multiple R-squared:  0.04165,    Adjusted R-squared:  0.04101
## F-statistic: 65.12 on 2 and 2997 DF,  p-value: < 2.2e-16
```

```
plot(Wage$age, Wage$wage)
```



Generate a sequence of age values spanning the range

```
agelims = Wage %>%
  select(age)%>%
  range
```

Get the min/max values of age using the range () function

```
grid1 = seq(from =min(agelims), to =max(agelims))
```

Fitting a regression spline using basic functions

```
fit = lm(wage~bs(age, df=6), data = Wage)
summary(fit)
```

```
##
## Call:
## lm(formula = wage ~ bs(age, df = 6), data = Wage)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -99.681 -24.403  -5.202  15.441 201.413
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)         56.314      7.258   7.759 1.17e-14 ***
## bs(age, df = 6)1    27.824     12.435   2.238   0.0253 *
## bs(age, df = 6)2    54.063      7.127   7.585 4.41e-14 ***
## bs(age, df = 6)3    65.828      8.323   7.909 3.62e-15 ***
## bs(age, df = 6)4    55.813      8.724   6.398 1.83e-10 ***
## bs(age, df = 6)5    72.131     13.745   5.248 1.65e-07 ***
## bs(age, df = 6)6    14.751     16.209   0.910   0.3629
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.91 on 2993 degrees of freedom
## Multiple R-squared:  0.08729,    Adjusted R-squared:  0.08546
## F-statistic: 47.71 on 6 and 2993 DF,  p-value: < 2.2e-16
```

```
pred = predict(fit, newdata = list(age = grid1), se = TRUE)

summary(pred)
```

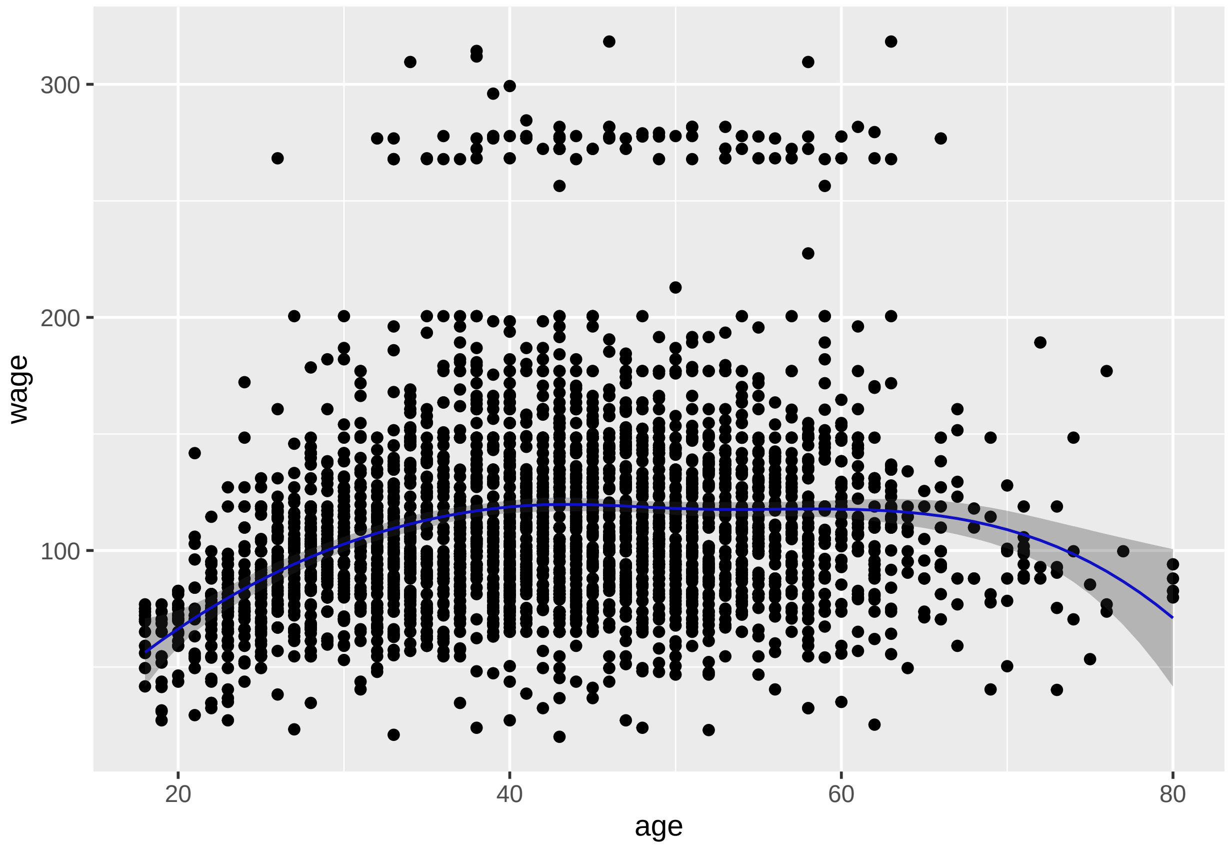
```
##                 Length Class  Mode
## fit             63     -none- numeric
## se.fit          63     -none- numeric
## df               1     -none- numeric
## residual.scale  1     -none- numeric
```

Compute error bands (2*SE)

```
se_bands = with(pred, cbind("upper" = fit+2*se.fit,
                            "lower" = fit-2*se.fit))
```

Plot the Spline and the error bands

```
ggplot() +
  geom_point(data = Wage, aes(x = age, y = wage)) +
  geom_line(aes(x = grid1, y = pred$fit), color = "#0000FF") +
  geom_ribbon(aes(x = grid1,
                  ymin = se_bands[,"lower"],
                  ymax = se_bands[,"upper"]),
              alpha = 0.3) +
  xlim(agelims)
```



Smooth Splines

Fit smooth splines

```
fit2 = with(Wage, smooth.spline(age, wage, df =20))
fit2_cv = with(Wage, smooth.spline(age, wage, cv = TRUE))
```

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

```
summary(fit2)
```

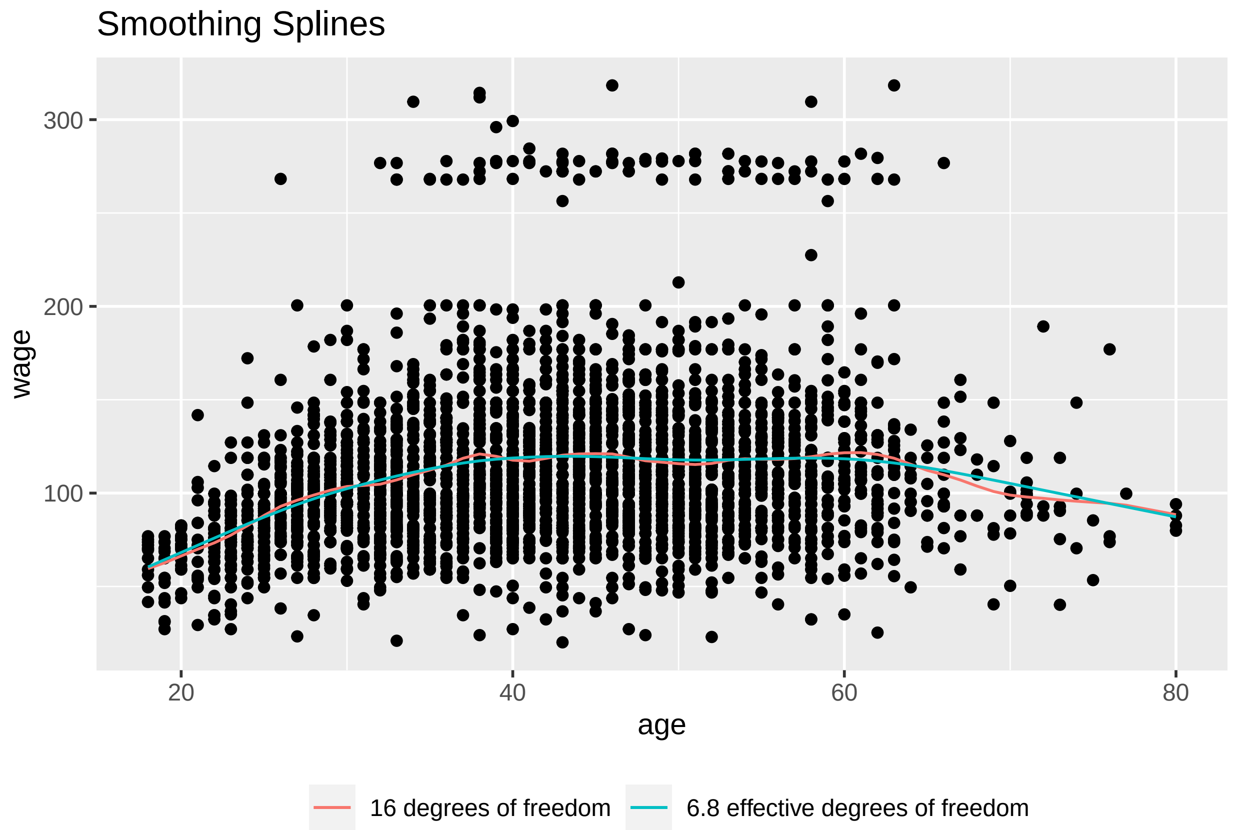
```
##            Length Class             Mode
## x          61     -none-            numeric
## y          61     -none-            numeric
## w          61     -none-            numeric
## yin        61     -none-            numeric
## tol         1     -none-            numeric
## data        3     -none-            list
## no.weights  1     -none-            logical
## lev        61     -none-            numeric
## cv.crit     1     -none-            numeric
## pen.crit    1     -none-            numeric
## crit        1     -none-            numeric
## df          1     -none-            numeric
## spar        1     -none-            numeric
## ratio       1     -none-            numeric
## lambda      1     -none-            numeric
## iparms      5     -none-            numeric
## auxM        0     -none-            NULL
## fit         5     smooth.spline.fit list
## call        4     -none-            call
```

```
summary(fit2_cv)
```

```
##            Length Class             Mode
## x          61     -none-            numeric
## y          61     -none-            numeric
## w          61     -none-            numeric
## yin        61     -none-            numeric
## tol         1     -none-            numeric
## data        3     -none-            list
## no.weights  1     -none-            logical
## lev        61     -none-            numeric
## cv.crit     1     -none-            numeric
## pen.crit    1     -none-            numeric
## crit        1     -none-            numeric
## df          1     -none-            numeric
## spar        1     -none-            numeric
## ratio       1     -none-            numeric
## lambda      1     -none-            numeric
## iparms      5     -none-            numeric
## auxM        0     -none-            NULL
## fit         5     smooth.spline.fit list
## call        4     -none-            call
```

Plot the smoothing splines

```
ggplot() +
  geom_point(data = Wage, aes(x = age, y = wage)) +
  geom_line(aes(x = fit2$x, y = fit2$y,
                color = "16 degrees of freedom"))  +
  geom_line(aes(x = fit2_cv$x, y = fit2_cv$y,
                color = "6.8 effective degrees of freedom")) +
  theme(legend.position = 'bottom')+
  labs(title = "Smoothing Splines", colour="")
```

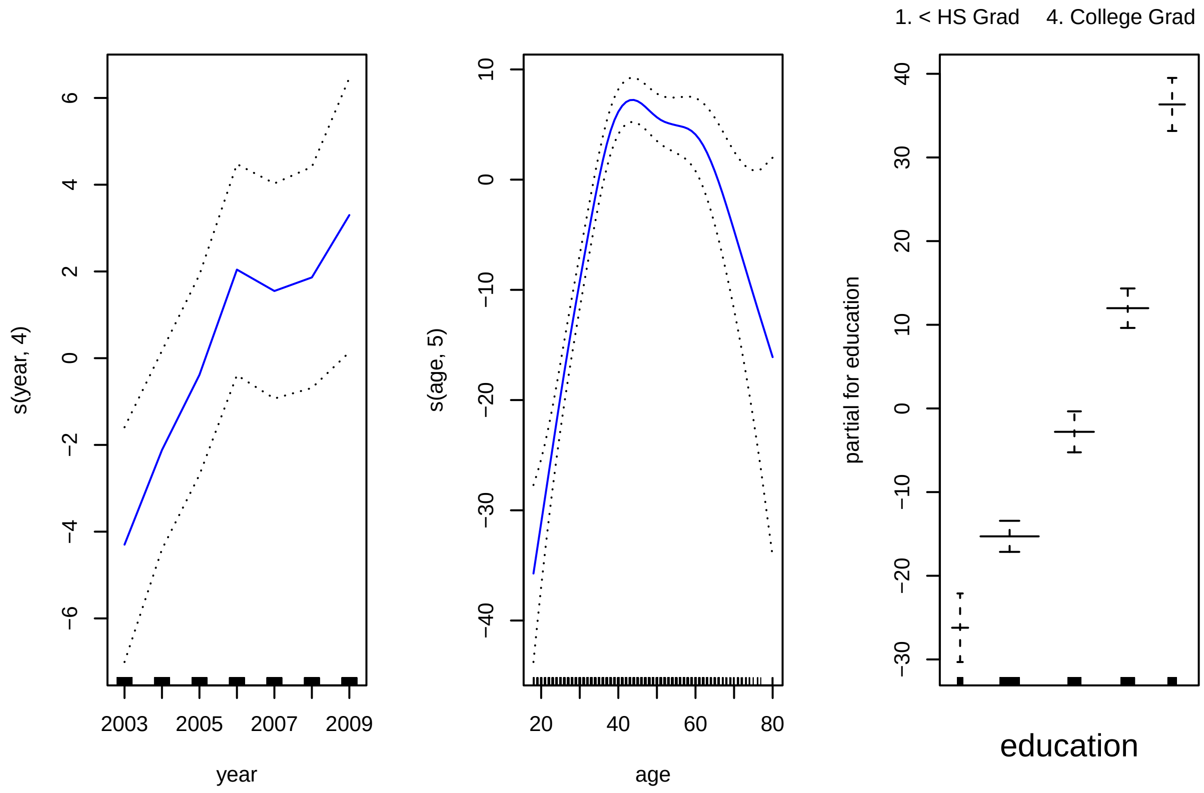


GAMs

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

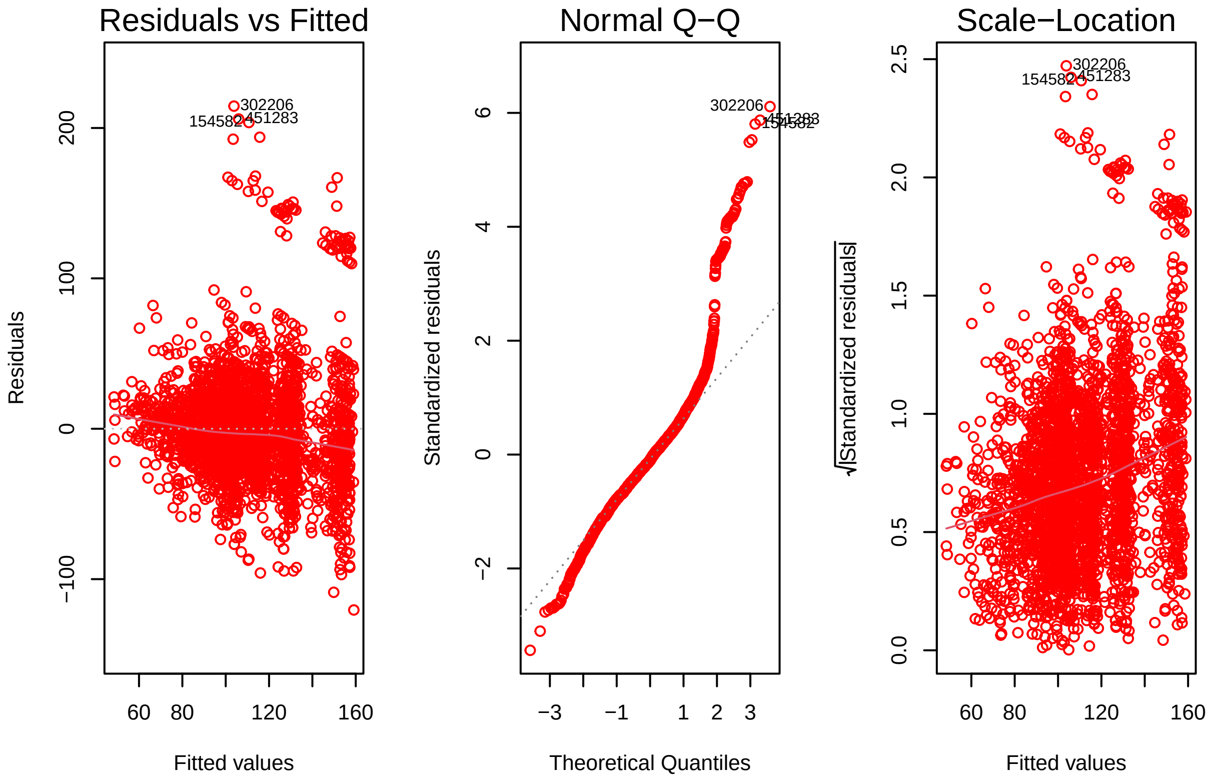
```
##
## Call:
## lm(formula = wage ~ ns(year, 4) + ns(age, 5) + education, data = Wage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -120.513  -19.608   -3.583   14.112  214.535
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     46.949      4.704   9.980  < 2e-16 ***
## ns(year, 4)1                     8.625      3.466   2.488  0.01289 *
## ns(year, 4)2                     3.762      2.959   1.271  0.20369
## ns(year, 4)3                    8.127      4.211   1.930  0.05375 .
## ns(year, 4)4                    6.806      2.397   2.840  0.00455 **
## ns(age, 5)1                    45.170      4.193  10.771  < 2e-16 ***
## ns(age, 5)2                    38.450      5.076   7.575 4.78e-14 ***
## ns(age, 5)3                    34.239      4.383   7.813 7.69e-15 ***
## ns(age, 5)4                    48.678     10.572   4.605 4.31e-06 ***
## ns(age, 5)5                     6.557      8.367   0.784  0.43328
## education2. HS Grad            10.983      2.430   4.520 6.43e-06 ***
## education3. Some College       23.473      2.562   9.163  < 2e-16 ***
## education4. College Grad       38.314      2.547  15.042  < 2e-16 ***
## education5. Advanced Degree    62.554      2.761  22.654  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35.16 on 2986 degrees of freedom
## Multiple R-squared:  0.293,  Adjusted R-squared:  0.2899
## F-statistic:  95.2 on 13 and 2986 DF,  p-value: < 2.2e-16
```

```
gam2 = gam(wage~s(year, 4) + s(age, 5) + education, data = Wage)
par(mfrow = c(1,3))
plot(gam2, se = TRUE, col = "blue")
```

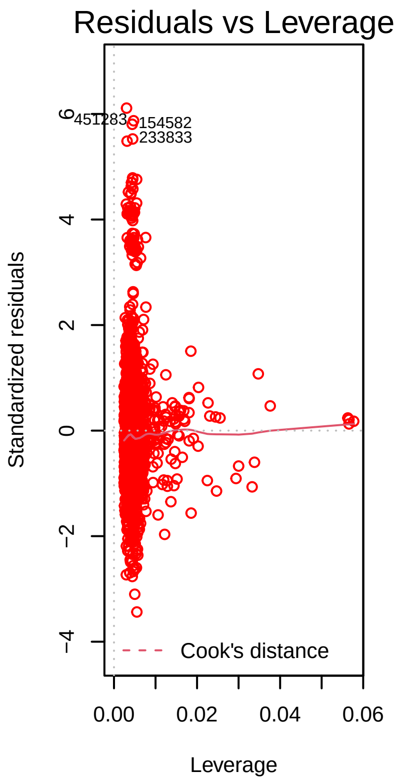


```
par(mfrow = c(1,3))
plot(gam1, se = TRUE, col = "red")
```

```
## Warning in plot.window(...): "se" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "se" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in box(...): "se" is not a graphical parameter

## Warning in title(...): "se" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "se" is not a graphical
## parameter

## Warning in plot.window(...): "se" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "se" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in box(...): "se" is not a graphical parameter

## Warning in title(...): "se" is not a graphical parameter

## Warning in plot.window(...): "se" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "se" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in box(...): "se" is not a graphical parameter

## Warning in title(...): "se" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "se" is not a graphical
## parameter
```



```
## Warning in plot.window(...): "se" is not a graphical parameter

## Warning in plot.xy(xy, type, ...): "se" is not a graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in axis(side = side, at = at, labels = labels, ...): "se" is not a
## graphical parameter

## Warning in box(...): "se" is not a graphical parameter

## Warning in title(...): "se" is not a graphical parameter

## Warning in plot.xy(xy.coords(x, y), type = type, ...): "se" is not a graphical
## parameter
```



```
gam_lm_year= gam(wage ~ year + s(age, 5) + education, data = Wage)
print(anova(gam_lm_year, gam2, test = "F"))
```

```
## Analysis of Deviance Table
##
## Model 1: wage ~ year + s(age, 5) + education
## Model 2: wage ~ s(year, 4) + s(age, 5) + education
##   Resid. Df Resid. Dev Df Deviance      F Pr(>F)
## 1      2989    3693842
## 2      2986    3689770  3   4071.1 1.0982 0.3486
```

```
summary(gam2)
```

```
##
## 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
```

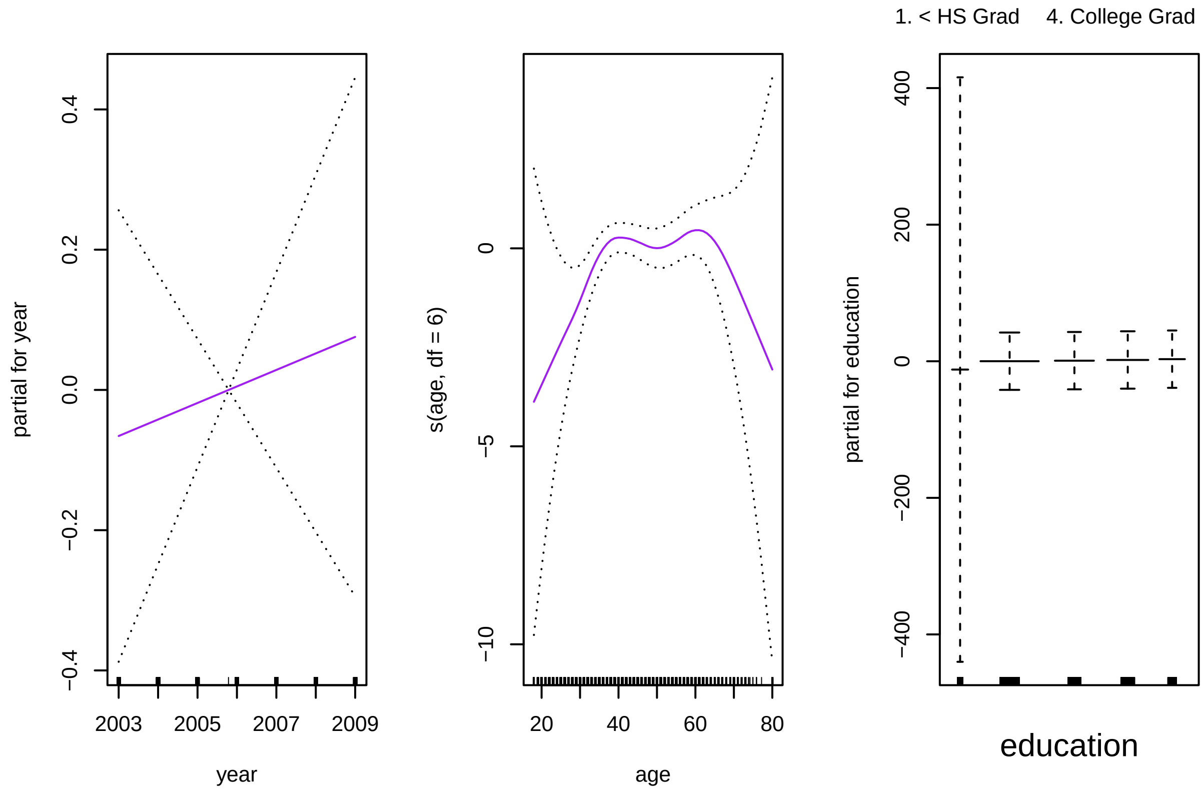
```
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##              Df  Sum Sq Mean Sq F value    Pr(>F)
## s(year, 4)    1   27162   27162  21.981 2.877e-06 ***
## s(age, 5)     1  195338  195338 158.081 < 2.2e-16 ***
## education     4 1069726  267432 216.423 < 2.2e-16 ***
## 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
```

```
preds = predict(gam_lm_year, newdata = Wage)
summary(preds)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   47.59   97.51  107.41  111.70  127.55  159.11
```

Logistic Regression GAMS

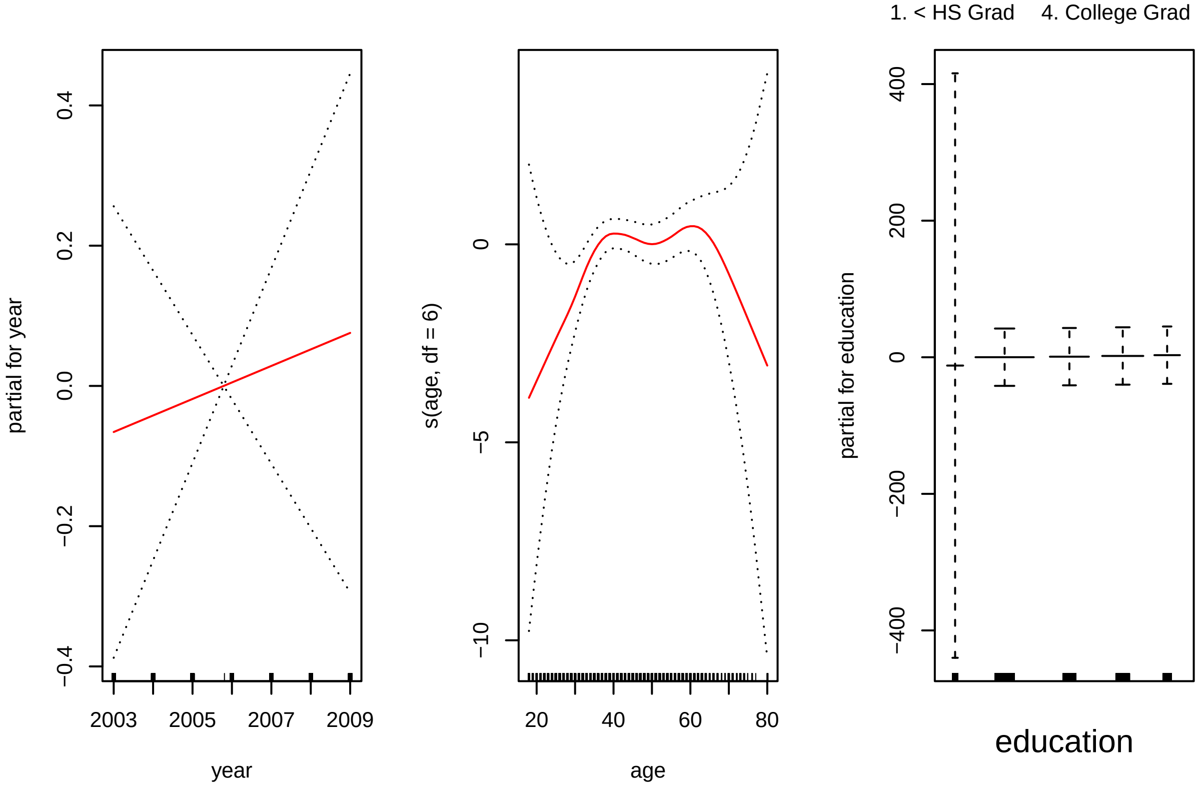
```
gam_log = gam(I(wage>250) ~ year + s(age, df = 6) + education,
              family = binomial, data = Wage)
par(mfrow=c(1,3))
plot(gam_log, se = TRUE, col = "purple")
```



```
with(Wage, 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
```

```
College_ed=Wage %>%
  filter(education !="1. <HS Grad")

gam_log2 = gam(I(wage>250) ~ year + s(age, df = 6) +education,
                family = binomial, data = College_ed)

par(mfrow=c(1,3))

plot(gam_log2, se = TRUE, col = "Red")
```


partial for year
0.4
0.2
0.0
−0.2
−0.4
2003
2005
2007
2009
year
s(age, df = 6)
0
−5
−10
20
40
60
80
age
1. < HS Grad
4. College Grad
partial for education
400
200
0
−200
−400
education