Homework 3

Yifang Zhang

Q1. Replicate Figure 7.11 using B-splines

a). Obtain the transformed data matrix for year and age:

```
library(ISLR)
#colnames(Wage)
#summary(Wage)
### a. obtain the transformed data matrix for year and age
library(splines)
ageM=bs(Wage$age,df=5, degree = 2,intercept = FALSE)
attr(ageM,"knots")
```

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

```
dim(ageM)
```

```
## [1] 3000    5
```

```
yearM=bs(Wage$year,df=4,degree = 1,intercept = FALSE)
attr(yearM, "knots")
```

```
##  25%  50%  75%
## 2004 2006 2008
```

```
dim(yearM)
```

```
## [1] 3000    4
```

year: To make the number of columns of the data matrix for year is 4, I used degree=1, which is actually a linear spline in the bs function with three uniform interior knots. According to the result, the three interior knots are 2004 (Q1), 2006 (median), and 2008 (Q3).

age: To make the number of columns of the data matrix for age is 5, I used degree=2, which is actually a quadratic spline in the bs function with three uniform interior knots as well. According to the result, the three interior knots are 33.75 (Q1), 42 (median), and 51 (Q3).

b). Construct the data matrix for f3:

```
### b. construct the data matrix for education
educationM=model.matrix(~education-1, data = Wage)[,-1]
dim(educationM)
```

```
## [1] 3000    4
```

The dimension of the data matrix is 3000×4.

c). Column-combine to obtain a big regression data matrix:

```
### c. column-combine and get the big regression data matrix
bMatrix=as.matrix(data.frame(matrix(1,nrow=3000),yearM,ageM,educationM))
colnames(bMatrix)=c("intercept","year.1","year.2","year.3","year.4","age.1","age.2",
                    "age.3","age.4","age.5","HS Grad", "Some College",
                    "College Grad", "Advanced Degree")
dim(bMatrix)
```

```
## [1] 3000   14
```

The dimension of the data matrix is 3000×14.

d). Fit the big regression and obtained the estimated coefficients:

```
### d. fit the big regression
fit.bs=lm(wage ~ bs(year,df=4,degree = 1,intercept = FALSE)
          +bs(age,df=5, degree = 2,intercept = FALSE)
          +education, data=Wage)
summary(fit.bs)
```

```
##
## Call:
## lm(formula = wage ~ bs(year, df = 4, degree = 1, intercept = FALSE) +
##     bs(age, df = 5, degree = 2, intercept = FALSE) + education,
##     data = Wage)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.924  -19.627   -3.643   14.136  214.224
##
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                        46.667      5.322   8.769
## bs(year, df = 4, degree = 1, intercept = FALSE)1    2.002      2.139   0.936
## bs(year, df = 4, degree = 1, intercept = FALSE)2    6.713      2.153   3.117
## bs(year, df = 4, degree = 1, intercept = FALSE)3    5.715      2.251   2.538
## bs(year, df = 4, degree = 1, intercept = FALSE)4    7.820      2.369   3.300
## bs(age, df = 5, degree = 2, intercept = FALSE)1    14.575      7.481   1.948
## bs(age, df = 5, degree = 2, intercept = FALSE)2    42.819      4.787   8.945
## bs(age, df = 5, degree = 2, intercept = FALSE)3    39.699      5.466   7.263
## bs(age, df = 5, degree = 2, intercept = FALSE)4    41.287      6.405   6.446
## bs(age, df = 5, degree = 2, intercept = FALSE)5    15.030     10.835   1.387
## education2. HS Grad                                10.962      2.430   4.511
## education3. Some College                           23.433      2.562   9.147
## education4. College Grad                           38.272      2.548  15.022
## education5. Advanced Degree                        62.518      2.761  22.641
##                                                  Pr(>|t|)
## (Intercept)                                       < 2e-16 ***
## bs(year, df = 4, degree = 1, intercept = FALSE)1 0.349467
## bs(year, df = 4, degree = 1, intercept = FALSE)2 0.001842 **
## bs(year, df = 4, degree = 1, intercept = FALSE)3 0.011195 *
## bs(year, df = 4, degree = 1, intercept = FALSE)4 0.000977 ***
## bs(age, df = 5, degree = 2, intercept = FALSE)1  0.051494 .
## bs(age, df = 5, degree = 2, intercept = FALSE)2   < 2e-16 ***
```

```
## bs(age, df = 5, degree = 2, intercept = FALSE)3  4.80e-13 ***
## bs(age, df = 5, degree = 2, intercept = FALSE)4  1.33e-10 ***
## bs(age, df = 5, degree = 2, intercept = FALSE)5  0.165485
## education2. HS Grad                              6.69e-06 ***
## education3. Some College                          < 2e-16 ***
## education4. College Grad                          < 2e-16 ***
## education5. Advanced Degree                       < 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
```

e). Compute the pointwise standard errors for f1,f2 and standard error for the estimates for different levels of education:

```
### e. compute the pointwise standard errors for f1, f2 on a grid
sigma=summary(fit.bs)$sigma
VarMatrix=solve(t(bMatrix)%*%bMatrix)*sigma^2
subM.year=VarMatrix[2:5,2:5]
yearLims=range(Wage$year)
year.grid=seq(from=yearLims[1], to=yearLims[2])
bsyeargrid=bs(year.grid,df=4,degree=1,intercept = FALSE)
pointstd.f1=sqrt(bsyeargrid%*%subM.year%*%t(bsyeargrid))

ageLims=range(Wage$age)
age.grid=seq(from=ageLims[1], to=ageLims[2])
subM.age=VarMatrix[6:10,6:10]
bsagegrid=bs(age.grid,df=5,degree=2, intercept = FALSE)
pointstd.f2=sqrt(bsagegrid%*%subM.age%*%t(bsagegrid))

subM.education=VarMatrix[c(1,11:14),c(1,11:14)]
stderr.education=sqrt(diag(subM.education))
stderr.education
```

```
##       intercept         HS Grad    Some College    College Grad Advanced Degree
##        5.321959        2.429778        2.561836        2.547726        2.761246
```

```
pointstd.f1
```

```
##      [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,]    0 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
## [2,]    0 1.426247 1.574292 1.154400 1.247070 1.283325 1.268013
## [3,]    0 1.574292 1.854125 1.696513 1.584369 1.534748 1.553651
## [4,]    0 1.154400 1.696513 2.153265 1.692668 1.457938 1.554967
## [5,]    0 1.247070 1.584369 1.692668 1.909949 1.871113 1.557113
## [6,]    0 1.283325 1.534748 1.457938 1.871113 1.988932 1.868140
## [7,]    0 1.268013 1.553651 1.554967 1.557113 1.868140 2.369380
```

```
View(pointstd.f2)
```

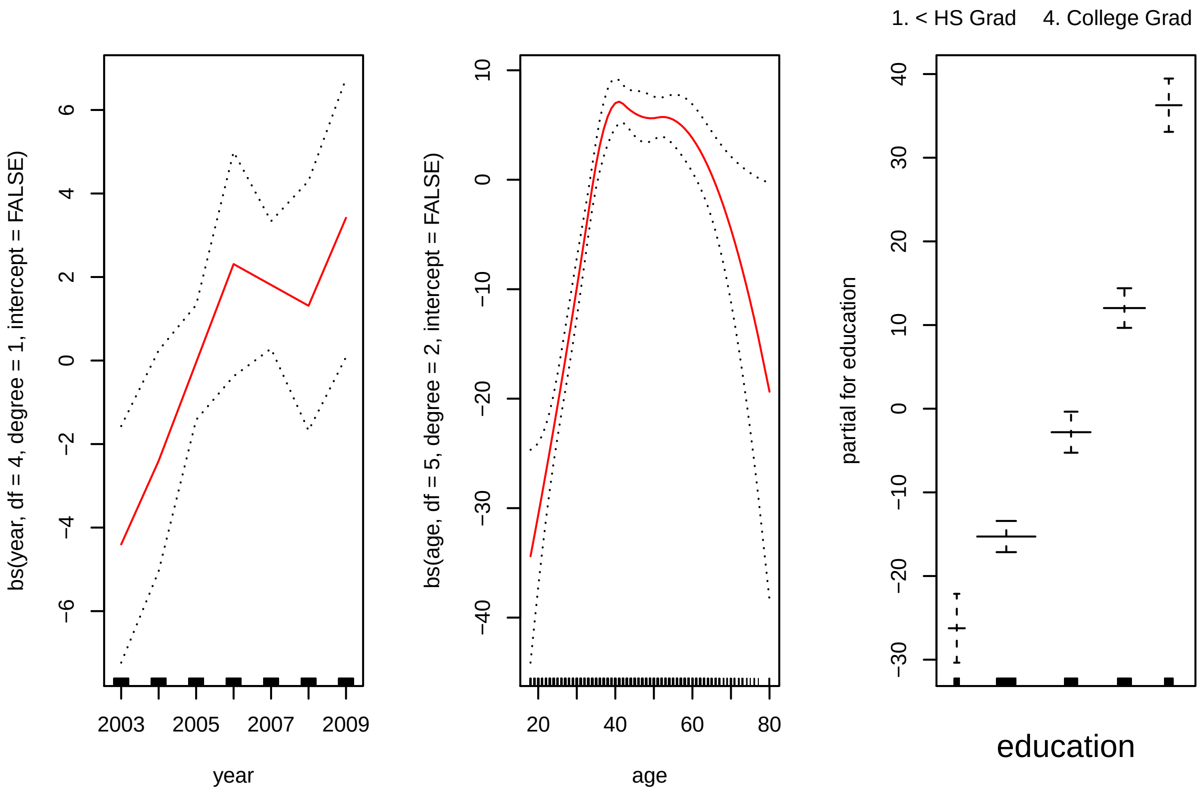
Here I created two sets of grid values for year and age respectively, and showed the result of pointwise standard error for f1 on that grid.

For education, the intercept is actually the coefficient for the base group (< HS Grad) and the standard error for the estimates for different levels of education is shown above as well.

f). Produce a similar graph as Figure 7.11:

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

```
## Loaded gam 1.20
```



Comparing the graph with Figure 7.11, we can find that the only difference is that Figure 7.11 has a wider ylim for both year and age. For example, in Figure 7.11, the ylim is from -30 to 30 while the ylim of my plot is just from -6 to 6. Except for that, this two graphs are almost the same.

Q2. Replicate Figure 7.11 using B-splines and backfitting approach

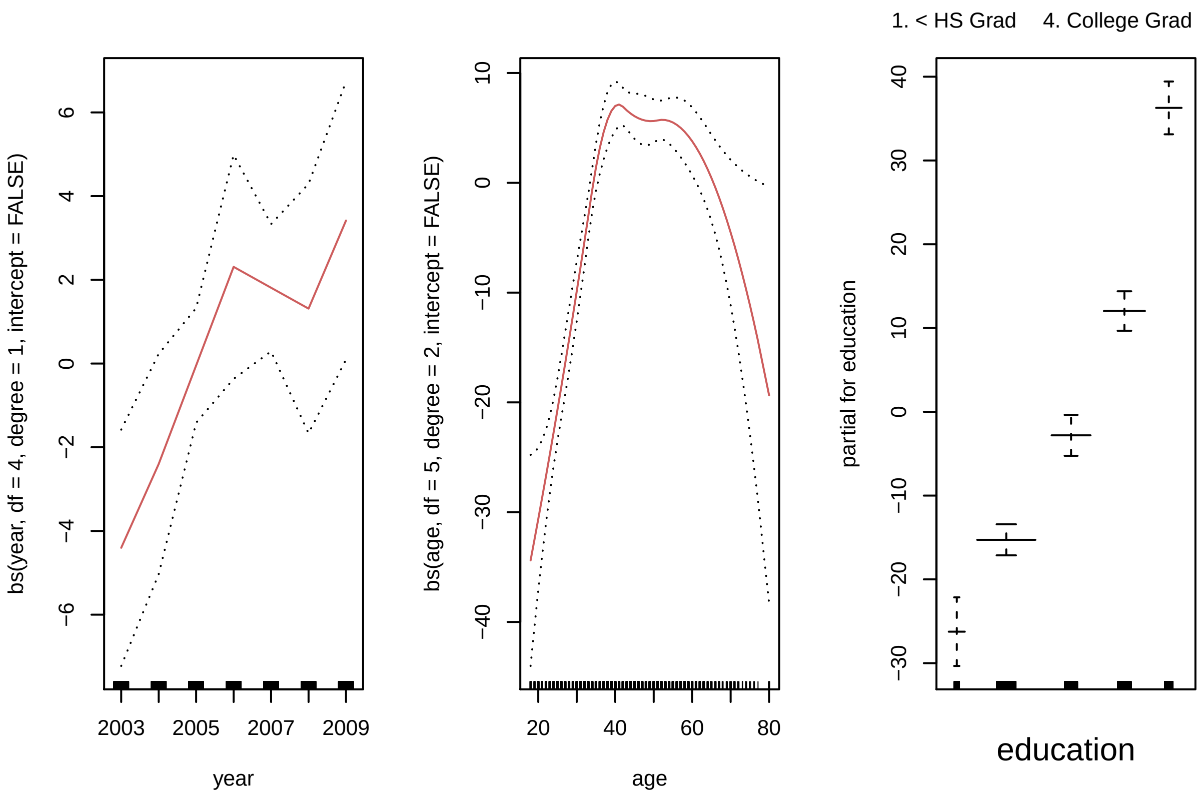
a). Write up the backfitting function and loop the backfitting approach for 100 times:

```
library(ISLR)
library(splines)
library(gam)

iter1=100
tol=1e-12
tol_curr=1
year=Wage$year
age=Wage$age
education=Wage$education
wage=Wage$wage
#initialize all fits as zeor
y=matrix(0,nrow=3000,ncol=1)
x1=matrix(0,nrow=3000,ncol=4)
x2=matrix(0,nrow=3000,ncol=5)
x3=matrix(0,nrow=3000,ncol=4)
f0=lm(y~1)
f1=lm(y ~x1)
f2=lm(y~x2)
f3=lm(y~x3)

#implement by residuals for 100 times or 1000 times
i=1
for (i in 1:iter1){
  f0.old=f0
  f1.old=f1
  f2.old=f2
  f3.old=f3
  M=cbind(fitted(f0),fitted(f1),fitted(f2),fitted(f3))
  newresponse=wage-fitted(f1.old)-fitted(f2.old)-fitted(f3.old)
  f0.old=lm(newresponse ~ 1)
  newresponse=wage-fitted(f0.old)-fitted(f2.old)-fitted(f3.old)
  f1.old=lm(newresponse ~ bs(year,df=4,degree = 1,intercept = FALSE))
  newresponse=wage-fitted(f0.old)-fitted(f1.old)-fitted(f3.old)
  f2.old=lm(newresponse ~ bs(age,df=5,degree=2,intercept = FALSE))
  newresponse=wage-fitted(f0.old)-fitted(f1.old)-fitted(f2.old)
  f3.old=lm(newresponse ~ education)
  M.old=cbind(fitted(f0.old),fitted(f1.old),fitted(f2.old),fitted(f3.old))
  tol_curr=sum(sqrt(colSums((M.old-M)^2)))/sum(sqrt((colSums(M^2))))
  if (tol_curr>tol){print(i)}
  f0=f0.old
  f1=f1.old
  f2=f2.old
  f3=f3.old
}
```

```
## [1] 1
## [1] 2
## [1] 3
```

```
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```

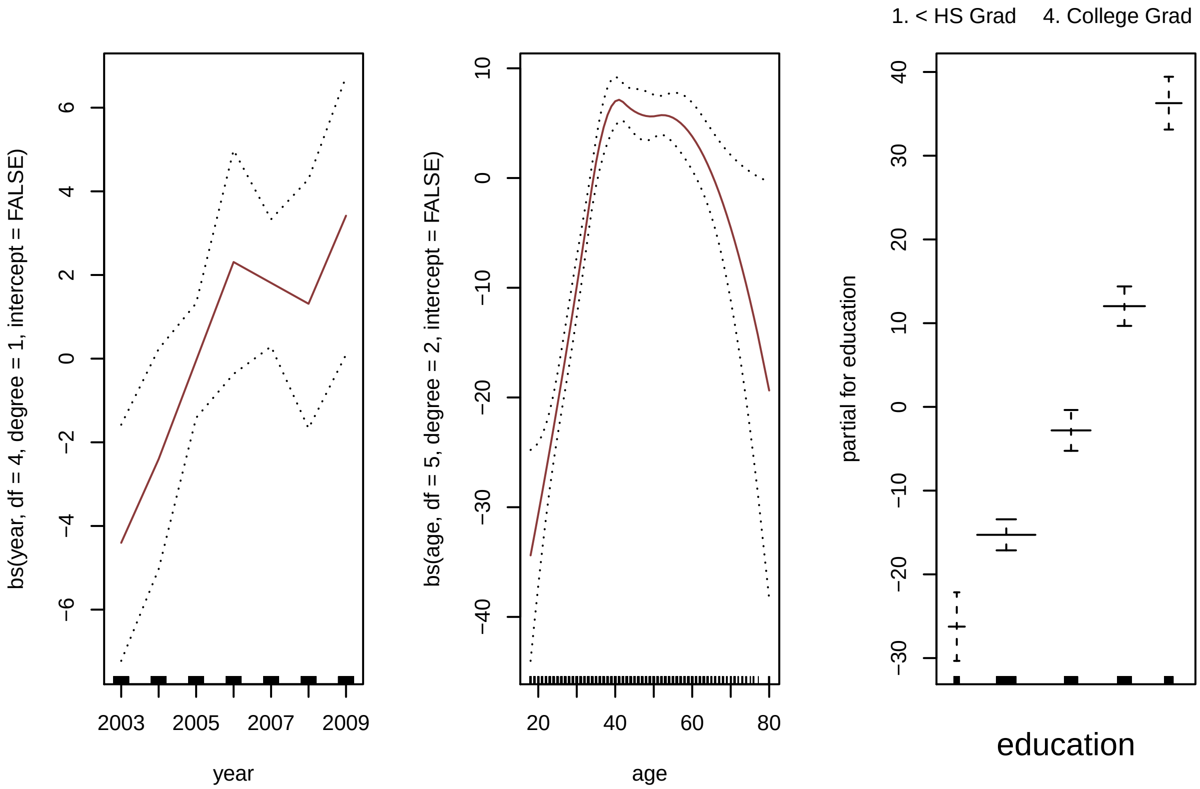
```r
#result shows that the fifth iteration can provide a good approximates
#summary(f0)
#summary(f1)
#summary(f2)
#summary(f3)

par(mfrow=c(1,3))
plot.Gam(f1,se=TRUE,col="indianred")
plot.Gam(f2,se=TRUE,col="indianred")
plot.Gam(f3,se=TRUE,col="indianred")
```



We can find that the new graph is exacly the same as what we obtained in Problem 1 (f).

b). Loop for 1000 times:

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
```



When the iteration times increased to 1000 times, we still get the exactly same graph comparing to the graph we obtained in Problem 1 (f).

c). How many iteration do we need to get a good approximates?

In part a) and b), the code I wrote will print each i where the current ratio after this i^{th} iteration is still larger than the stop criterion 10^{-12} we set at the beginning. And if we check the result, we will find that just after 9 iteration, we can obtain a good approximates to what we obtained in Problem 1 (f).

Q3. Using backfitting for the additive logistic regression model.

a). Obtain the transformed data matrix:

```
library(ISLR)
library(splines)
library(gam)
Wage.new=subset(Wage, education != "1. < HS Grad")
### a. the transformed data matrix for age using B-Splines
year.new=Wage.new$year
age.new=Wage.new$age
ageM.new=bs(age.new,df=5,degree=2,intercept = FALSE)
dim(ageM.new)
```

```
## [1] 2732    5
```

```
attr(ageM.new,"knots")
```

```
## 25% 50% 75%
##  34  42  51
```

Again, I used three interior knots and put degree=2 to get the transformed data matrix for age.

b). Construct the data matrix for f3:

```
### b. the data matrix for f3
education.new=Wage.new$education
educationM.new=model.matrix(~education-1, data=Wage.new)[,-c(1,2)]
dim(educationM.new)
```

```
## [1] 2732    3
```

c). Get the big regression data matrix:

```
### c. Column-combine
bMatrix.new=as.matrix(data.frame(matrix(1,nrow=dim(ageM.new)[1]),
                                 year.new,ageM.new,educationM.new))
dim(bMatrix.new)
```

```
## [1] 2732   10
```

The dimension of this matrix is 2732×10 since we remove data points whose education level is less than a high school education.

d) and e). Fit the additive logistic model using IRLS:

```
### d. fit the additive logistic model using IRLS
dummywage=matrix(0,nrow=length(Wage.new$wage))
for (k in 1:length(dummywage)){
  if (Wage.new$wage[k] > 250){dummywage[k]=1}
}
wage.new=Wage.new$wage
### new y is the dummy variable of wage
yhat=mean(dummywage)
m=length(dummywage)
delta=1e-12
```

```
iter=100
alpha=log(yhat/(1-yhat))
beta=c(alpha,rep(0,ncol(bMatrix.new)-1))
currtol=1
i=0
X=bMatrix.new
while (currtol>delta && i<iter){
  i=i+1
  beta_old=beta
  eta=X%*%beta_old
  p=plogis(eta)
  weight=p*(1-p)
  W=matrix(0,nrow=m,ncol=m)
  for (k in 1:m){
    W[k,k]=weight[k]
  }
  z=eta+(dummywage-p)/weight
  beta_old=solve(t(X)%*%W%*%X)%*%(t(X)%*%(W%*%z))
  diff=beta_old-beta
  numerator=sqrt(sum((diff[2]*X[,2])^2))+sqrt(sum(((X[,3:7]%*%diff[3:7]))^2))+sqrt(sum(((X[,8:10]%*%dif
  denom=sqrt(sum((beta[2]*X[,2])^2))+sqrt(sum(((X[,3:7]%*%beta[3:7]))^2))+sqrt(sum(((X[,8:10]%*%beta[8:
  currtol=numerator/denom
  beta=beta_old
}
rownames(beta)=c("intercept","year","age.1","age.2","age.3","age.4","age.5",
                 "Some College","College Grad", "Advanced Degree")
beta
```

```
##                         [,1]
## intercept       -56.18403548
## year              0.02384616
## age.1             0.54231156
## age.2             3.88064848
## age.3             2.93022493
## age.4             4.45462754
## age.5             0.62525810
## Some College      0.78357792
## College Grad      1.82655432
## Advanced Degree   3.00027652
```

```
Weight.final=W
```

After fitting the data using the IRLS agorithm, the estimated coefficients are shown above.

f). Obtain pointwise standard errors for year and age, and standard error for education:

```
### e. Compute std error
CovM=solve(t(bMatrix.new)%*%W%*%bMatrix.new)
yearLims.new=range(Wage.new$year)
year.grid=seq(from=yearLims.new[1], to=yearLims.new[2])
hyear=as.matrix(year.grid)
f1.irls.std=sqrt(hyear%*%CovM[2,2]%*%t(hyear))
f1.irls.std
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]
## [1,] 115.9541 115.9830 116.0119 116.0409 116.0698 116.0987 116.1276
## [2,] 115.9830 116.0119 116.0409 116.0698 116.0987 116.1277 116.1566
## [3,] 116.0119 116.0409 116.0698 116.0988 116.1277 116.1566 116.1856
## [4,] 116.0409 116.0698 116.0988 116.1277 116.1567 116.1856 116.2145
## [5,] 116.0698 116.0987 116.1277 116.1567 116.1856 116.2146 116.2435
## [6,] 116.0987 116.1277 116.1566 116.1856 116.2146 116.2435 116.2724
## [7,] 116.1276 116.1566 116.1856 116.2145 116.2435 116.2724 116.3014
```

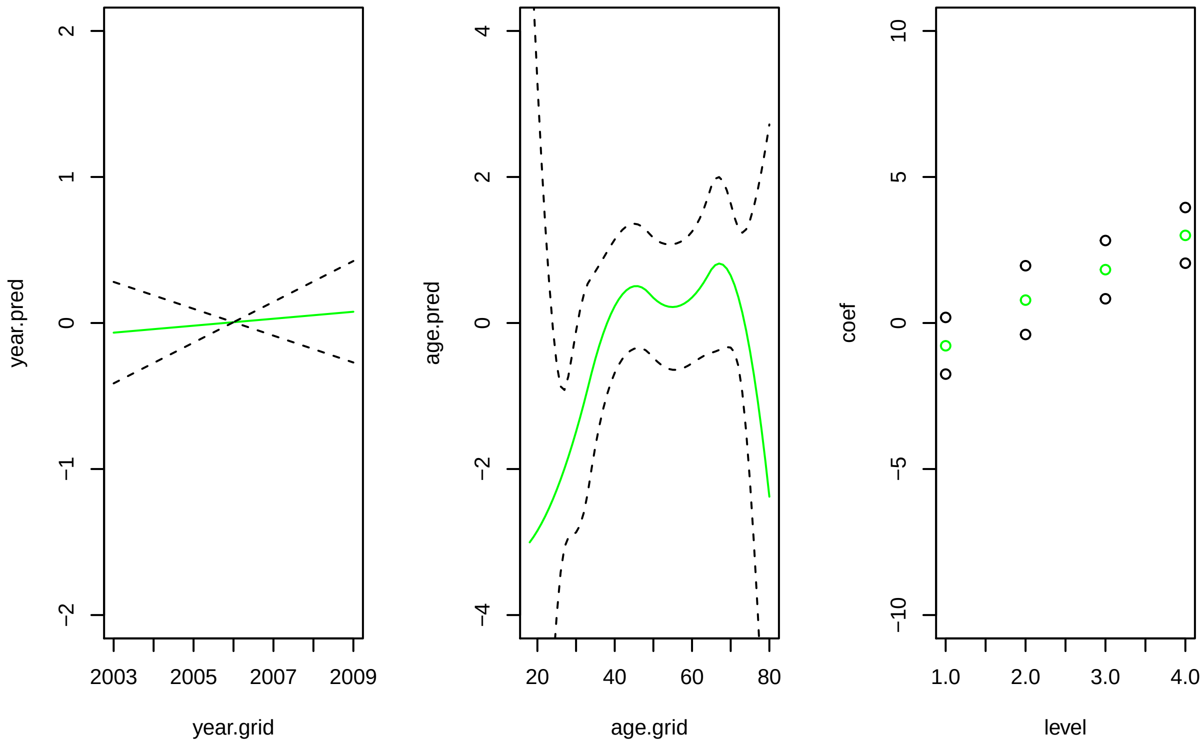
```
ageLims.new=range(Wage.new$age)
age.grid=seq(from=ageLims.new[1], to=ageLims.new[2])
hage=bs(age.grid,df=5,degree=2,intercept = FALSE)
f2.irls.std=sqrt(hage%*%CovM[3:7,3:7]%*%t(hage))
View(f2.irls.std)

education.irls.std=rep(0,4)
for (i in 1:4){
  if (i==1){education.irls.std[i]=sqrt(CovM[1,1])}
  else {education.irls.std[i]=sqrt(CovM[i+6,i+6])}
}
education.irls.std
```

```
## [1] 116.1568980   0.5886856   0.4987915   0.4764680
```

g). Plot figures:

```
### f. obtain the graph
par(mfrow=c(1,3))
year.pred=beta[2]*year.grid-47.83
newhyear=as.matrix(year.grid-2006)
stdnew.year=sqrt(CovM[2,2])*sqrt(diag(newhyear%*%t(newhyear)))
plot(year.grid,year.pred,type="l",col="green", ylim = c(-2,2))
lines(year.grid,year.pred+2*stdnew.year,lty=2)
lines(year.grid,year.pred-2*stdnew.year,lty=2)

newhage=bs(age.grid,df=5,degree=2,intercept = FALSE)
age.pred=newhage%*%beta[3:7]-3.0044
hnew.age=scale(bs(age.grid,df=5, degree=2,intercept = FALSE),scale=FALSE)
stdnew.age=sqrt(diag(hnew.age%*%CovM[3:7,3:7]%*%t(hnew.age)))
plot(age.grid,age.pred,type="l",col="green",ylim = c(-4,4))
lines(age.grid,age.pred-2*stdnew.age,lty=2)
lines(age.grid,age.pred+2*stdnew.age,lty=2)

BIGM=as.matrix(data.frame(matrix(1,nrow=dim(ageM.new)[1]),
                          scale(year.new,scale=FALSE),
                          scale(ageM.new,scale = FALSE),educationM.new))
CovM.new=solve(t(BIGM)%*%W%*%BIGM)
stdnew.education=rep(0,4)
for (i in 1:4){
  if (i==1){stdnew.education[i]=sqrt(CovM.new[i,i])}
  else {stdnew.education[i]=sqrt(CovM.new[i+6,i+6])}
}
level=c(1,2,3,4)
coef=c(-0.7799,beta[8:10])
```

```
lower=coef-2*stdnew.education
upper=coef+2*stdnew.education
plot(level,coef,type="p",col="green", ylim=c(-10,10))
lines(level,lower,lty=2, type="p")
lines(level,upper,lty=2,type="p")
```



For part (f), we centered our data matrix and get the above figure and fixed the prediction of year,age and education by using the intercept information I got when I first applied the backfitting algorithm.

The results shown that it is very close to the Figure 7.14.

h) and i). Using weighted backfitting:

```
### h&i. IRLS & f. plot a figure similar to Figure 7.14
## set initial intercept and f1,f2,f3
alpha.irls=log(yhat/(1-yhat))
n=length(dummywage)
y.irls=matrix(0,nrow=n,ncol=1)
x1.irls=matrix(0,nrow=n,ncol=1)
x2.irls=matrix(0,nrow=n,ncol=5)
x3.irls=matrix(0,nrow=n,ncol=3)
#f0.irls=lm(y.irls~1)
f1.irls=lm(y.irls ~x1.irls)
f2.irls=lm(y.irls~x2.irls)
f3.irls=lm(y.irls~x3.irls)

#implement by residuals for 100 times or 1000 times
j=1
itertimes=100
currtol.irls=1
```

```r
delta.irls=1e-12

for (j in 1:itertimes){
  alpha.old=alpha.irls
  f1.old=f1.irls
  f2.old=f2.irls
  f3.old=f3.irls
  M=cbind(fitted(f1.irls),fitted(f2.irls),fitted(f3.irls))
  eta=alpha.irls+fitted(f1.old)+fitted(f2.old)+fitted(f3.old)
  p=plogis(eta)
  weight=p*(1-p)
  z=eta+(dummywage-p)/(p*(1-p))

  newresponse=z-fitted(f1.old)-fitted(f2.old)-fitted(f3.old)
  f0.old=lm(newresponse~1,weights = weight)
  alpha.old=f0.old$coefficients
  newresponse=z-fitted(f0.old)-fitted(f2.old)-fitted(f3.old)
  f1.old=lm(newresponse ~ year.new, weights = weight)
  newresponse=z-fitted(f0.old)-fitted(f1.old)-fitted(f3.old)
  f2.old=lm(newresponse ~ bs(age.new,df=5,degree=2,intercept = FALSE),weights = weight)
  newresponse=z-fitted(f0.old)-fitted(f1.old)-fitted(f2.old)
  f3.old=lm(newresponse ~ education.new,weights = weight)

  M.old=cbind(fitted(f1.old),fitted(f2.old),fitted(f3.old))
  currtol.irls=sum(sqrt(colSums((M.old-M)^2)))/sum(sqrt((colSums(M^2))))
  if (currtol.irls>delta.irls){print(j)}

  alpha.irls=alpha.old
  f1.irls=f1.old
  f2.irls=f2.old
  f3.irls=f3.old
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

```r
summary(f1.irls)
```

```
##
## Call:
## lm(formula = newresponse ~ year.new, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4236 -0.1916 -0.0946 -0.0730 16.6973
##
## Coefficients:
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -47.82962  109.62531  -0.436    0.663
## year.new      0.02385    0.05465   0.436    0.663
##
## Residual standard error: 0.948 on 2730 degrees of freedom
## Multiple R-squared:  6.973e-05,  Adjusted R-squared:  -0.0002965
## F-statistic: 0.1904 on 1 and 2730 DF,  p-value: 0.6626
```

```
summary(f2.irls)
```

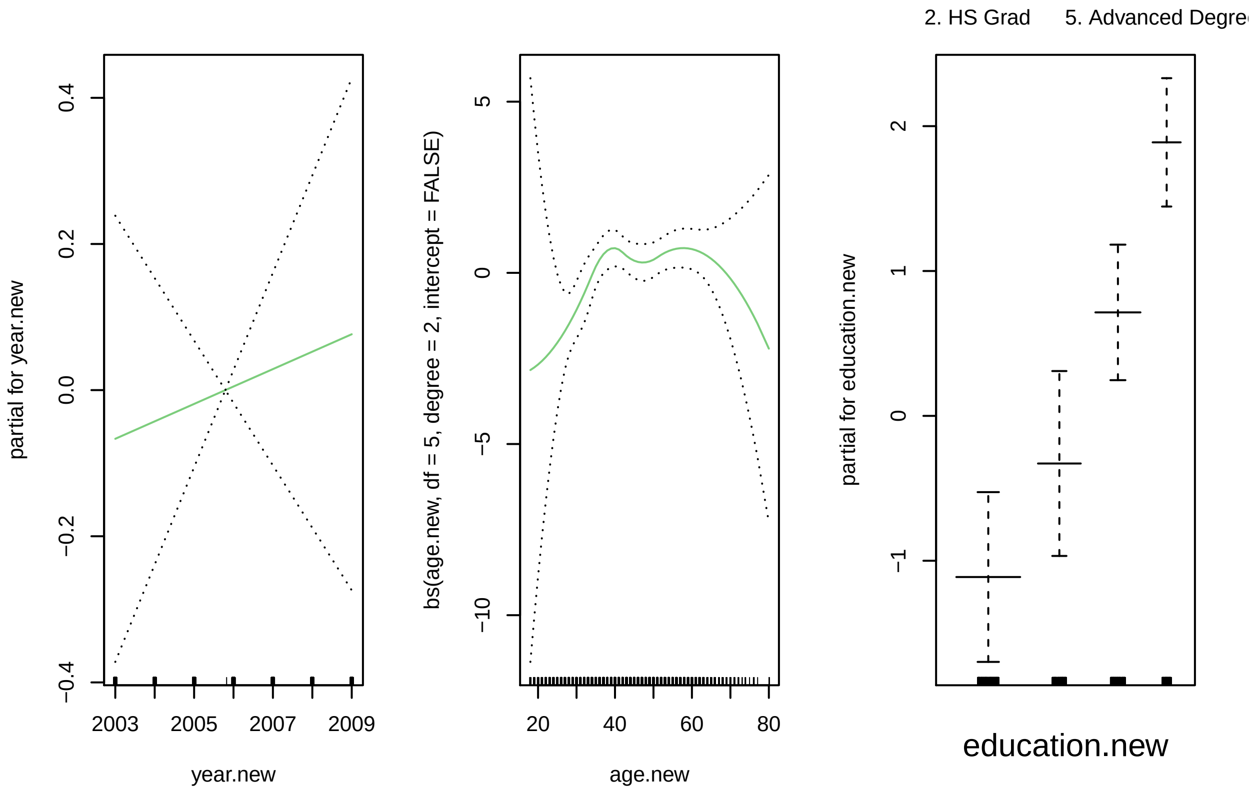
```
##
## Call:
## lm(formula = newresponse ~ bs(age.new, df = 5, degree = 2, intercept = FALSE),
##     weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4236 -0.1916 -0.0946 -0.0730 16.6973
##
## Coefficients:
##                                                     Estimate Std. Error t value
## (Intercept)                                          -3.0044     4.4226  -0.679
## bs(age.new, df = 5, degree = 2, intercept = FALSE)1   0.5423     5.1135   0.106
## bs(age.new, df = 5, degree = 2, intercept = FALSE)2   3.8806     4.3381   0.895
## bs(age.new, df = 5, degree = 2, intercept = FALSE)3   2.9302     4.4763   0.655
## bs(age.new, df = 5, degree = 2, intercept = FALSE)4   4.4546     4.4250   1.007
## bs(age.new, df = 5, degree = 2, intercept = FALSE)5   0.6253     5.2017   0.120
##                                                     Pr(>|t|)
## (Intercept)                                            0.497
## bs(age.new, df = 5, degree = 2, intercept = FALSE)1    0.916
## bs(age.new, df = 5, degree = 2, intercept = FALSE)2    0.371
## bs(age.new, df = 5, degree = 2, intercept = FALSE)3    0.513
## bs(age.new, df = 5, degree = 2, intercept = FALSE)4    0.314
## bs(age.new, df = 5, degree = 2, intercept = FALSE)5    0.904
##
## Residual standard error: 0.9487 on 2726 degrees of freedom
## Multiple R-squared:  0.004655,   Adjusted R-squared:  0.002829
## F-statistic:  2.55 on 5 and 2726 DF,  p-value: 0.02608
```

```
summary(f3.irls)
```

```
##
## Call:
## lm(formula = newresponse ~ education.new, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4236 -0.1916 -0.0946 -0.0730 16.6973
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -0.7799     0.4255  -1.833 0.066917 .
## education.new3. Some College      0.7836     0.5580   1.404 0.160362
## education.new4. College Grad      1.8266     0.4729   3.863 0.000115 ***
## education.new5. Advanced Degree   3.0003     0.4514   6.647  3.6e-11 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9483 on 2728 degrees of freedom
## Multiple R-squared:  0.02623,    Adjusted R-squared:  0.02516
## F-statistic:  24.5 on 3 and 2728 DF,  p-value: 1.223e-15
```

```r
par(mfrow=c(1,3))
plot.Gam(f1.irls,se=TRUE,col="palegreen3")
plot.Gam(f2.irls,se=TRUE,col="palegreen3")
plot.Gam(f3.irls,se=TRUE,col="palegreen3")
```



First, we can find that after 11 iterations we can actually get a good approximates comparing to what we obtained in part e). And the figure is quite close to Figure 7.14 and what we got in part (f).