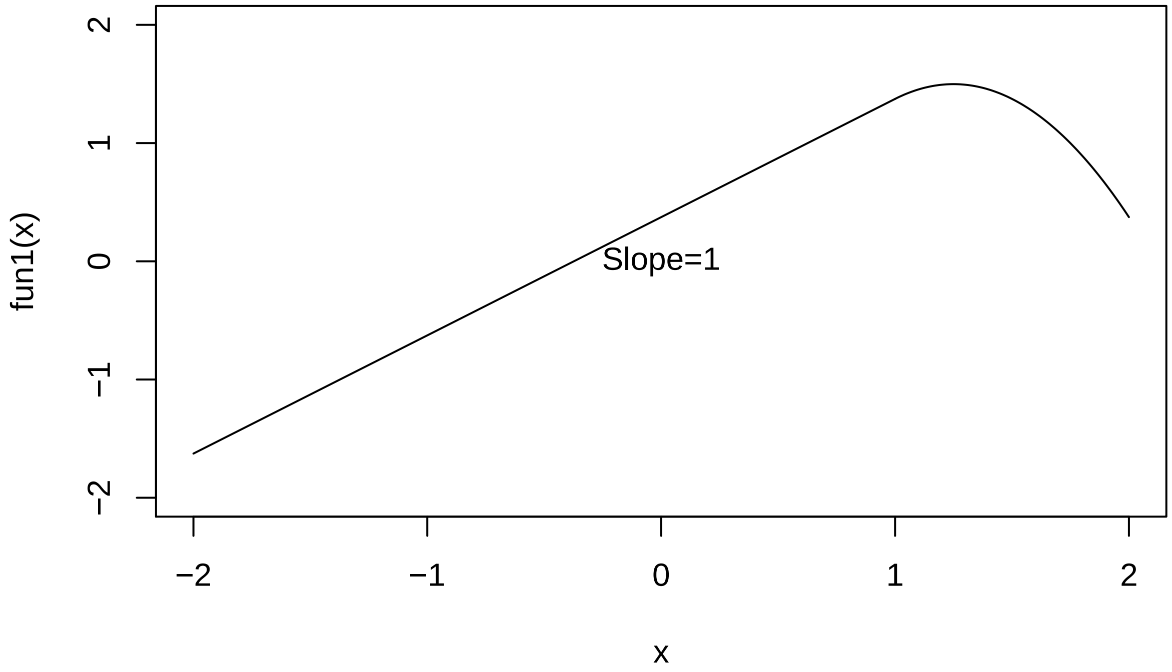
GAM Homework

Jiachen Feng

2021/2/25

7.3

```r
fun1 <- function(x){
  set.seed(1)
  1+x+rnorm(1)
}

fun2 <- function(x){
  set.seed(1)
  1+x-2*(x-1)^2+rnorm(1)
}

curve(fun1,-2,1,xlim = c(-2,2),ylim = c(-2,2))
curve(fun2,1,2,add = TRUE)
text(0,0,c("Slope=1"))
```



7.9

(a)

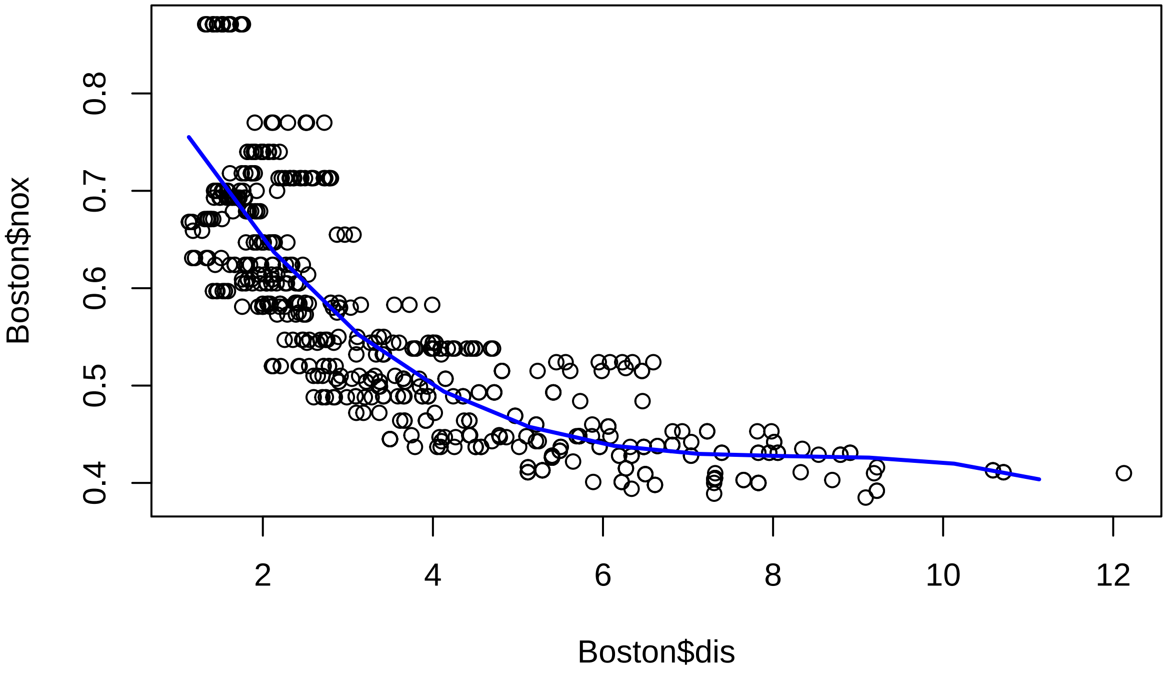
```
library(MASS)
```

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

```
data("Boston")
fit <- lm(nox~poly(dis,3),data = Boston)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
dis.grid <- seq(from=range(Boston$dis)[1],to=range(Boston$dis)[2])
preds <- predict(fit,newdata = list(dis=dis.grid),se=T)

plot(Boston$dis,Boston$nox)
lines(dis.grid,preds$fit,lwd=2,col="blue")
```



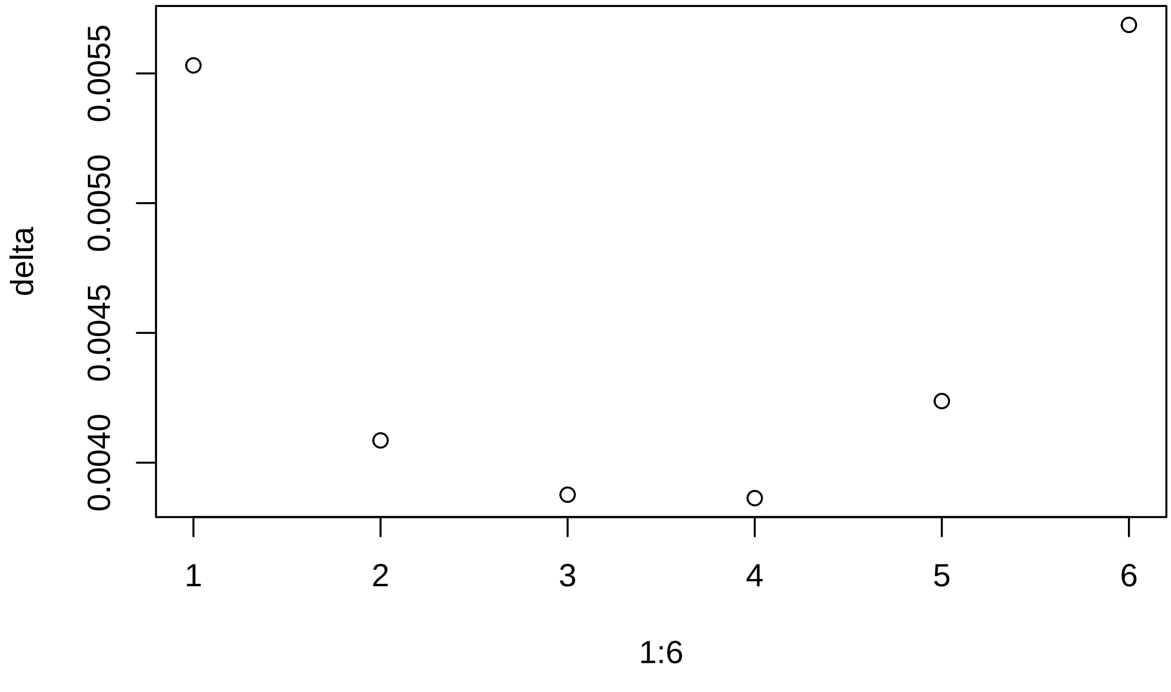
(b)

```
fit.1 <- lm(nox~dis,data = Boston)
fit.2 <- lm(nox~poly(dis,2),data = Boston)
fit.3 <- lm(nox~poly(dis,3),data = Boston)
fit.4 <- lm(nox~poly(dis,4),data = Boston)
fit.5 <- lm(nox~poly(dis,5),data = Boston)
fit.6 <- lm(nox~poly(dis,6),data = Boston)
anova(fit.1,fit.2,fit.3,fit.4,fit.5,fit.6)
```

```
## Analysis of Variance Table
##
## Model 1: nox ~ dis
## Model 2: nox ~ poly(dis, 2)
## Model 3: nox ~ poly(dis, 3)
## Model 4: nox ~ poly(dis, 4)
## Model 5: nox ~ poly(dis, 5)
## Model 6: nox ~ poly(dis, 6)
##   Res.Df    RSS Df Sum of Sq        F    Pr(>F)
## 1    504 2.7686
## 2    503 2.0353  1   0.73330 194.8174 < 2.2e-16 ***
## 3    502 1.9341  1   0.10116  26.8741  3.16e-07 ***
## 4    501 1.9330  1   0.00113   0.2990   0.58477
## 5    500 1.9153  1   0.01769   4.7001   0.03063 *
## 6    499 1.8783  1   0.03703   9.8385   0.00181 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c)

```r
library(boot)
delta <- rep(NA,6)
for (i in 1:6) {
  fit <- glm(nox~poly(dis,i),data = Boston)
  delta[i] <- cv.glm(Boston,fit,K=10)$delta[1]

}
plot(1:6,delta)
```



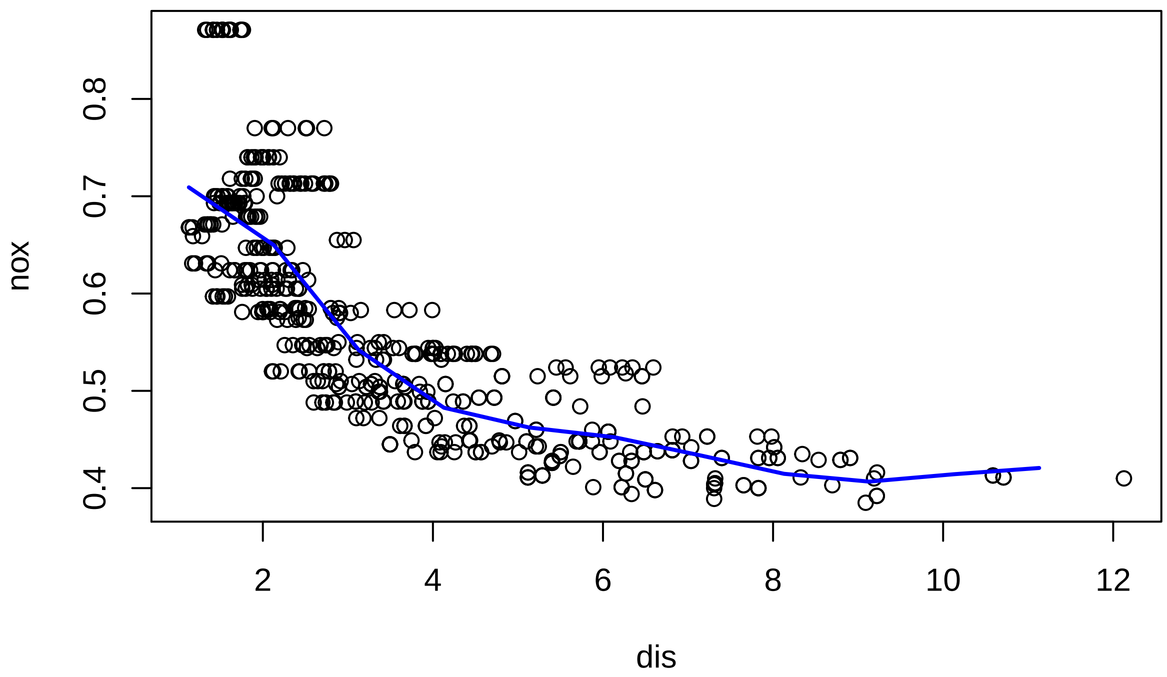
(d)

```r
library(splines)
#4 degrees of freedom
#The range of dis is about 0 to 12
```

```
fit <- lm(nox~bs(dis,knots = c(3,6,9)),data = Boston)
summary(fit)
```

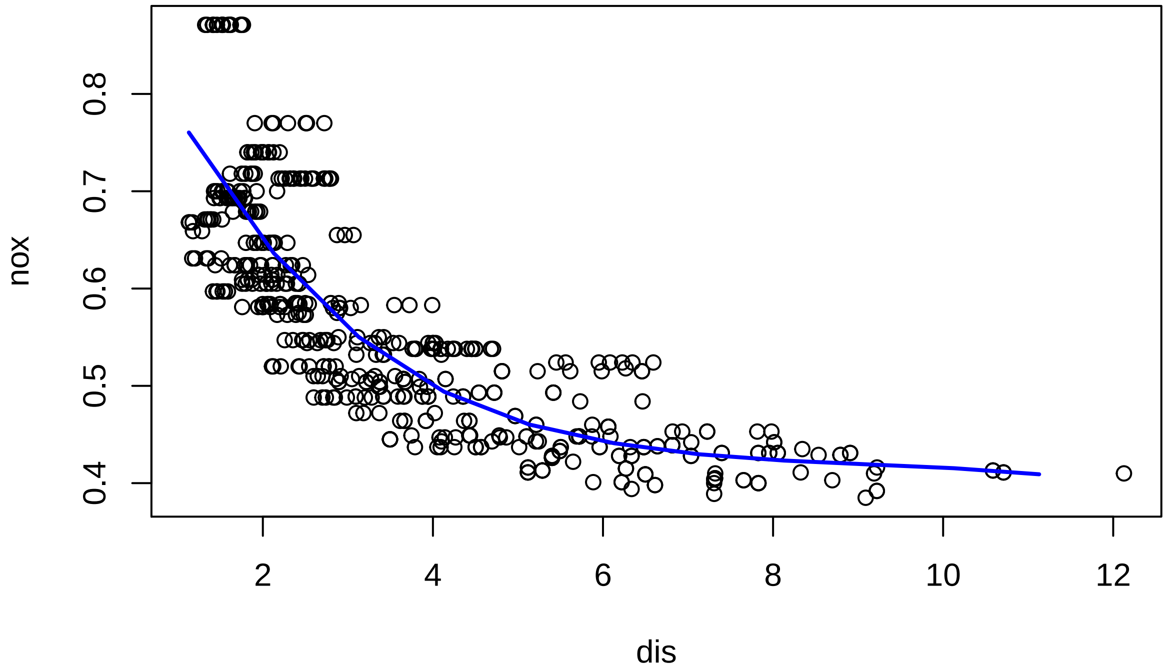
```
##
## Call:
## lm(formula = nox ~ bs(dis, knots = c(3, 6, 9)), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132134 -0.039466 -0.009042  0.025344  0.187258
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    0.709144   0.016099  44.049  < 2e-16 ***
## bs(dis, knots = c(3, 6, 9))1  0.006631   0.025467   0.260    0.795
## bs(dis, knots = c(3, 6, 9))2 -0.258296   0.017759 -14.544  < 2e-16 ***
## bs(dis, knots = c(3, 6, 9))3 -0.233326   0.027248  -8.563  < 2e-16 ***
## bs(dis, knots = c(3, 6, 9))4 -0.336530   0.032140 -10.471  < 2e-16 ***
## bs(dis, knots = c(3, 6, 9))5 -0.269575   0.058799  -4.585 5.75e-06 ***
## bs(dis, knots = c(3, 6, 9))6 -0.303386   0.062631  -4.844 1.70e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0612 on 499 degrees of freedom
## Multiple R-squared:  0.7244, Adjusted R-squared:  0.7211
## F-statistic: 218.6 on 6 and 499 DF,  p-value: < 2.2e-16
```

```
preds <- predict(fit,newdata = list(dis=dis.grid),se=T)
plot(nox~dis,data = Boston)
lines(dis.grid,preds$fit,lwd=2,col="blue")
```

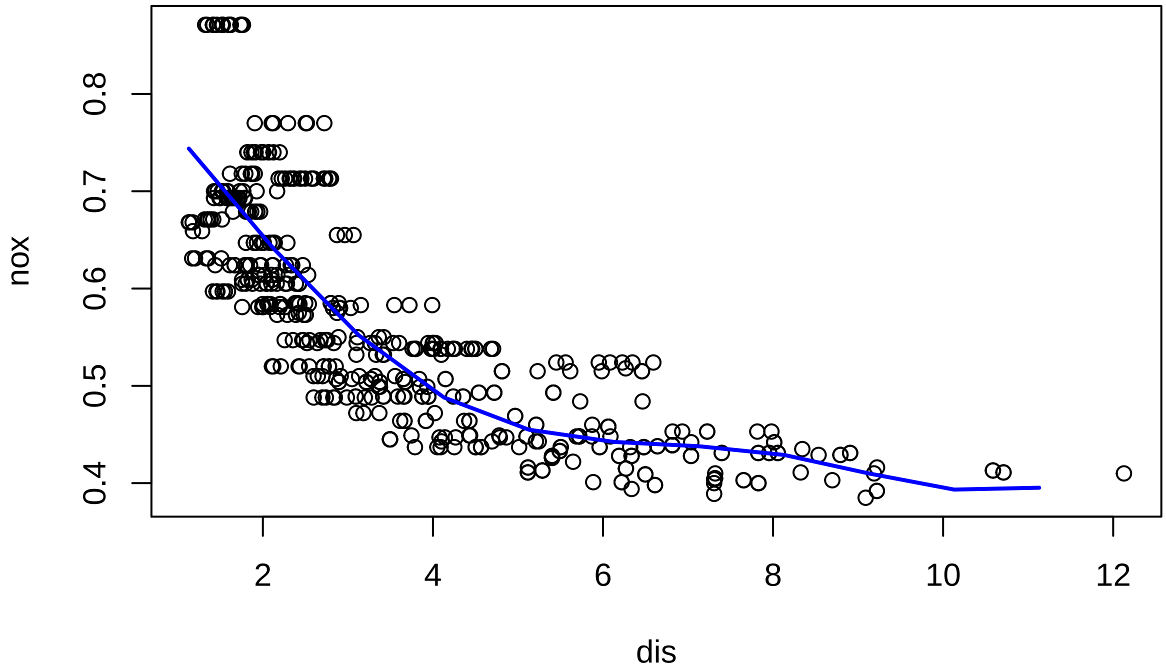


(e)

```
#The range of dis is about 0 to 12
fit.1 <- lm(nox~bs(dis,knots =6),data = Boston)

preds <- predict(fit.1,newdata = list(dis=dis.grid),se=T)
plot(nox~dis,data = Boston)
lines(dis.grid,preds$fit,lwd=2,col="blue")
```



```
fit.2 <- lm(nox~bs(dis,knots =c(4,8)),data = Boston)

preds <- predict(fit.2,newdata = list(dis=dis.grid),se=T)
plot(nox~dis,data = Boston)
lines(dis.grid,preds$fit,lwd=2,col="blue")
```



```
anova(fit.1,fit.2)
```

```
## Analysis of Variance Table
##
## Model 1: nox ~ bs(dis, knots = 6)
## Model 2: nox ~ bs(dis, knots = c(4, 8))
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    501 1.9311
## 2    500 1.9186  1  0.012542 3.2685 0.07123 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(f)

```
library(boot)
delta <- rep(NA,2)
fit.1 <- glm(nox~bs(dis,knots =6),data = Boston)
delta[1] <- cv.glm(Boston,fit.1,K=10)$delta[1]
```

```
## Warning in bs(dis, degree = 3L, knots = 6, Boundary.knots = c(1.137, 12.1265:
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = 6, Boundary.knots = c(1.137, 12.1265:
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = 6, Boundary.knots = c(1.1296, 10.7103:
## some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(dis, degree = 3L, knots = 6, Boundary.knots = c(1.1296, 10.7103:
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
fit.2 <- glm(nox~bs(dis,knots =c(4,8)),data = Boston)
delta[2] <- cv.glm(Boston,fit.2,K=10)$delta[2]
```

```
## Warning in bs(dis, degree = 3L, knots = c(4, 8), Boundary.knots = c(1.1296, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(4, 8), Boundary.knots = c(1.1296, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(4, 8), Boundary.knots = c(1.137, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(4, 8), Boundary.knots = c(1.137, :
## some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
delta
```

```
## [1] 0.003866883 0.003928526
```

7.10

(a)

```
data("College")

library(leaps)
```

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

```
regfit <- regsubsets(College$Outstate~.,data = College,method = "forward")
summary(regfit)
```

```
## Subset selection object
## Call: regsubsets.formula(College$Outstate ~ ., data = College, method = "forward")
## 17 Variables  (and intercept)
##             Forced in Forced out
## PrivateYes      FALSE      FALSE
## Apps            FALSE      FALSE
## Accept          FALSE      FALSE
## Enroll          FALSE      FALSE
## Top10perc       FALSE      FALSE
## Top25perc       FALSE      FALSE
## F.Undergrad     FALSE      FALSE
## P.Undergrad     FALSE      FALSE
## Room.Board      FALSE      FALSE
## Books           FALSE      FALSE
## Personal        FALSE      FALSE
## PhD             FALSE      FALSE
## Terminal        FALSE      FALSE
## S.F.Ratio       FALSE      FALSE
## perc.alumni     FALSE      FALSE
## Expend          FALSE      FALSE
## Grad.Rate       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: forward
##          PrivateYes Apps Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        " "  " "    " "    " "       " "       " "        
## 2  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 3  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 4  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 5  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 6  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 7  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
## 8  ( 1 ) "*"        " "  " "    " "    " "       " "       " "        
##          P.Undergrad Room.Board Books Personal PhD Terminal S.F.Ratio
## 1  ( 1 ) " "         " "        " "   " "      " " " "      " "      
## 2  ( 1 ) " "         " "        " "   " "      " " " "      " "      
## 3  ( 1 ) " "         "*"        " "   " "      " " " "      " "      
## 4  ( 1 ) " "         "*"        " "   " "      " " " "      " "      
## 5  ( 1 ) " "         "*"        " "   " "      "*" " "      " "      
## 6  ( 1 ) " "         "*"        " "   " "      "*" " "      " "      
## 7  ( 1 ) " "         "*"        " "   "*"      "*" " "      " "      
## 8  ( 1 ) " "         "*"        " "   "*"      "*" "*"      " "      
##          perc.alumni Expend Grad.Rate
## 1  ( 1 ) " "         "*"    " "      
## 2  ( 1 ) " "         "*"    " "      
## 3  ( 1 ) " "         "*"    " "      
## 4  ( 1 ) "*"         "*"    " "      
## 5  ( 1 ) "*"         "*"    " "      
## 6  ( 1 ) "*"         "*"    "*"      
## 7  ( 1 ) "*"         "*"    "*"      
## 8  ( 1 ) "*"         "*"    "*"      
```

```
coef(regfit,8)
```

```
##   (Intercept)    PrivateYes    Room.Board      Personal           PhD 
## -3427.0217976  2751.0006770     0.9097872    -0.3216904    18.8340453 
##      Terminal   perc.alumni        Expend     Grad.Rate 
##    24.7935526    43.3961323     0.2220780    28.3705072 
```

(b)

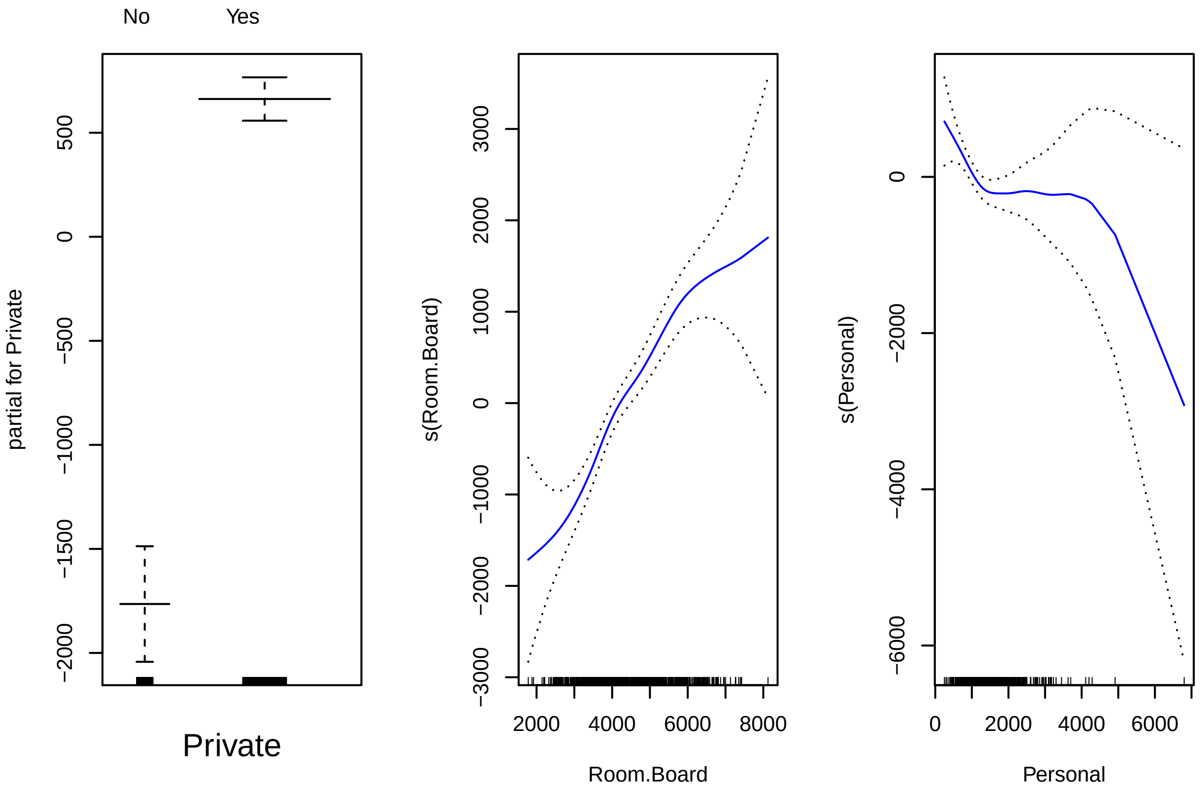
```
names(coef(regfit,8))
```

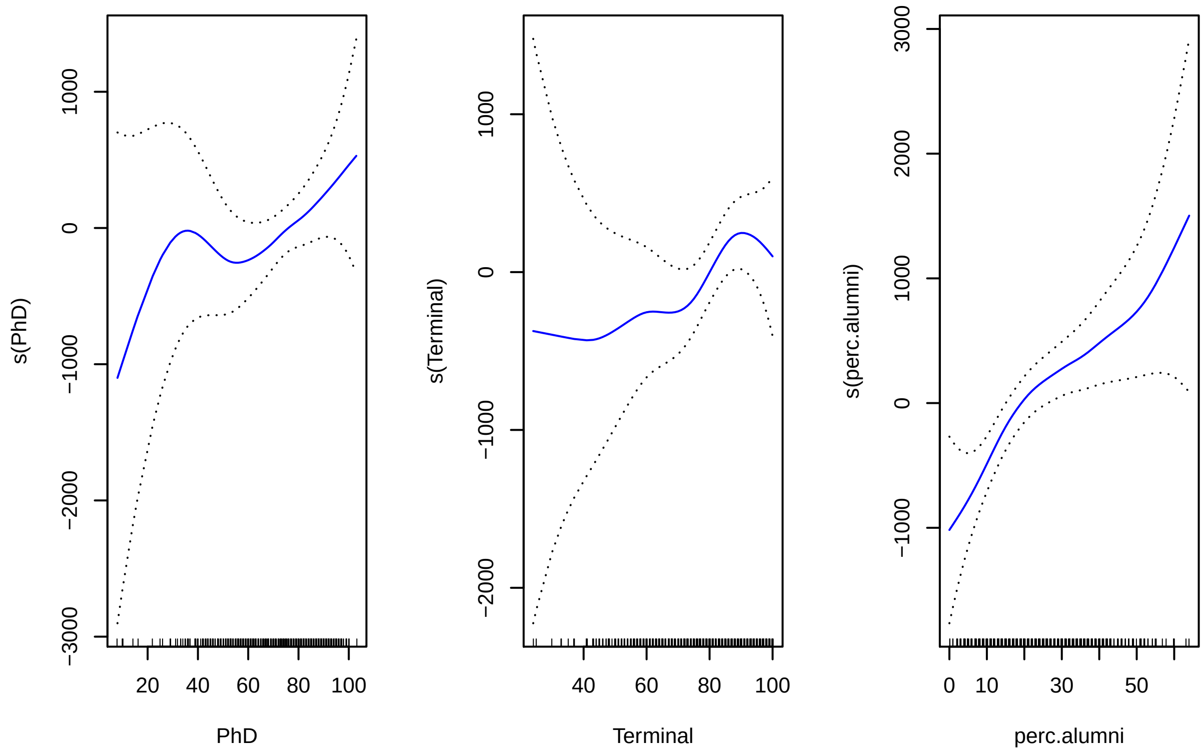
```
## [1] "(Intercept)" "PrivateYes"  "Room.Board"  "Personal"    "PhD"        
## [6] "Terminal"    "perc.alumni" "Expend"      "Grad.Rate"  
```

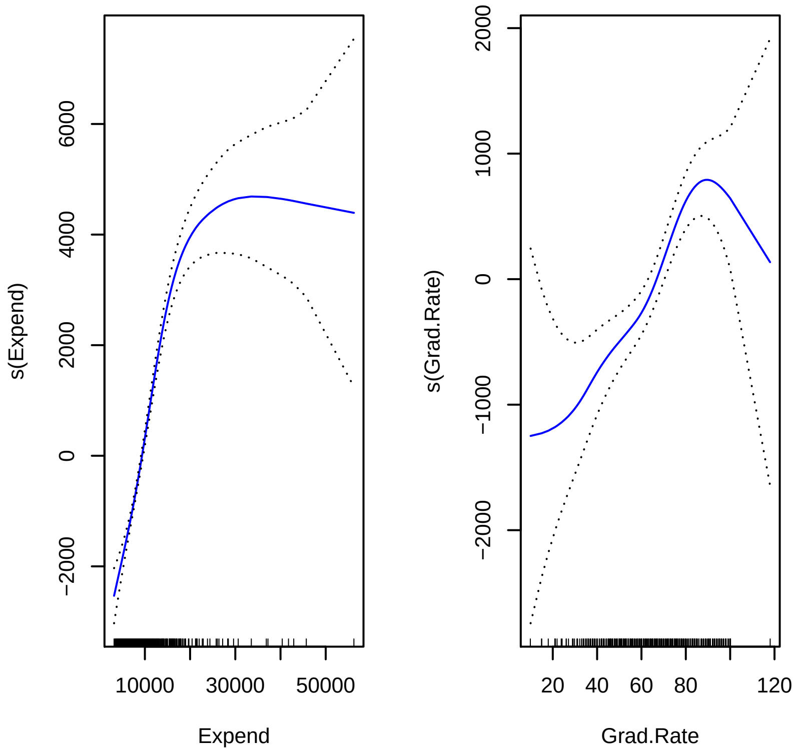
```
library(gam)
```

```
## 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 <- gam(Outstate~Private+s(Room.Board)+s(Personal)+s(PhD)+s(Terminal)+s(perc.alumni)+s(Expend)+s(Grad

par(mfrow=c(1,3))
plot(gam, se = T, col = "blue")
```


No
Yes
partial for Private
500
0
−500
−1000
−1500
−2000
Private
s(Room.Board)
3000
2000
1000
0
−1000
−2000
−3000
2000
4000
6000
8000
Room.Board
s(Personal)
0
−2000
−4000
−6000
0
2000
4000
6000
Personal


s(PhD)
1000
0
−1000
−2000
−3000
20
40
60
80
100
PhD
s(Terminal)
1000
0
−1000
−2000
40
60
80
100
Terminal
s(perc.alumni)
3000
2000
1000
0
−1000
0
10
30
50
perc.alumni



(c)

```r
summary(gam)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board) + s(Personal) +
##     s(PhD) + s(Terminal) + s(perc.alumni) + s(Expend) + s(Grad.Rate),
##     data = College)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7337.10 -1082.06    50.97  1309.61  7331.90
##
## (Dispersion Parameter for gaussian family taken to be 3393220)
##
##     Null Deviance: 12559297426 on 776 degrees of freedom
## Residual Deviance: 2534736579 on 747.0003 degrees of freedom
## AIC: 13920.41
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                 Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private          1 3383077692 3383077692 997.011 < 2.2e-16 ***
## s(Room.Board)    1 2516993731 2516993731 741.771 < 2.2e-16 ***
## s(Personal)      1   73897857   73897857  21.778 3.627e-06 ***
## s(PhD)           1  822522120  822522120 242.402 < 2.2e-16 ***
## s(Terminal)      1   55722569   55722569  16.422 5.601e-05 ***
## s(perc.alumni)   1  430666539  430666539 126.920 < 2.2e-16 ***
## s(Expend)        1 1044198084 1044198084 307.731 < 2.2e-16 ***
## s(Grad.Rate)     1  130940347  130940347  38.589 8.684e-10 ***
## Residuals      747 2534736579    3393220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                Npar Df Npar F   Pr(F)
## (Intercept)
## Private
## s(Room.Board)        3  2.301 0.07600 .
## s(Personal)          3  2.987 0.03044 *
## s(PhD)               3  1.966 0.11769
## s(Terminal)          3  1.644 0.17777
## s(perc.alumni)       3  1.209 0.30547
## s(Expend)            3 39.191 < 2e-16 ***
## s(Grad.Rate)         3  3.029 0.02879 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(d) Personal, Expend and Grad.Rate.

7.11

(a)

```
set.seed(100)
y <- rnorm(100,mean=10)
x1 <- rnorm(100,mean=1,sd=.5)
x2 <- rnorm(100,mean=100,sd=10)
dat <- data.frame(y,x1,x2)
```

(b)

```
beta1 <- 1
```

(c)

```
a <- y-beta1*x1
beta2 <- lm(a~x2)$coef[2]
```

(d)

```
a <- y-beta2*x2
beta1 <- lm(a~x1)$coef[2]
```

(e)

```
# library(tidyverse)
# beta <- data.frame(0,beta1,beta2)
# colnames(beta)[1] <- 'beta0'
# for (i in 1:1000) {
# a <- y-beta1*x1
# beta2 <- lm(a~x2)$coef[2]
# a <- y-beta2*x2
# beta1 <- lm(a~x1)$coef[2]
# beta0 <- lm(a~x1)$coef[1]
# add_row(beta0,beta1,beta2)
# }
```