P8106 HW 2

Minjie Bao

Contents

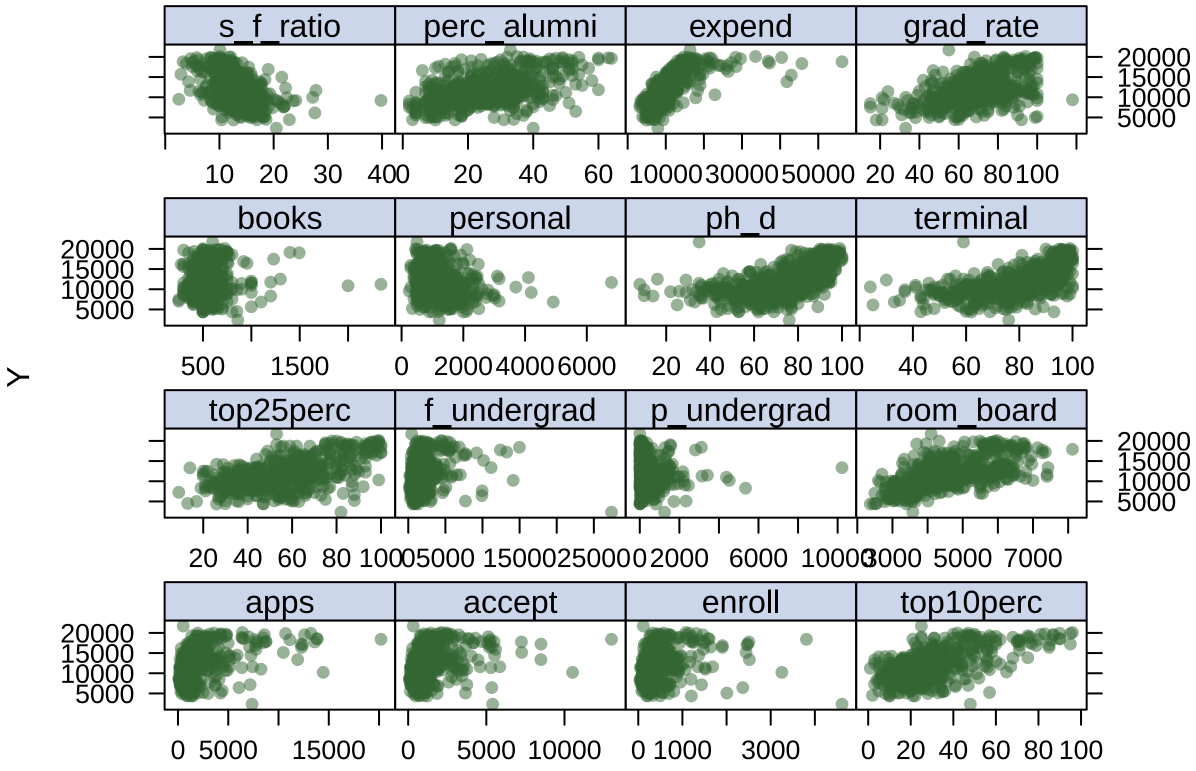
```
library(caret)
library(splines)
library(mgcv)
library(pdp)
library(earth)
library(tidyverse)
library(ggplot2)

set.seed(2021)
```

Data preparation

```
college_df = read_csv(’./data/College.csv’) %>%
  janitor::clean_names()

  #skimr::skim(college_df)

college_train = college_df %>%
  filter(college != "Columbia University") %>%
  select(-college)
```

a) Perform exploratory data analysis

```
# matrix of predictors
x = model.matrix(outstate ~ ., college_train)[,-1]
# vector of response
y = college_train$outstate

theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("","Y"),
            type = c("p"), layout = c(4, 4))
```

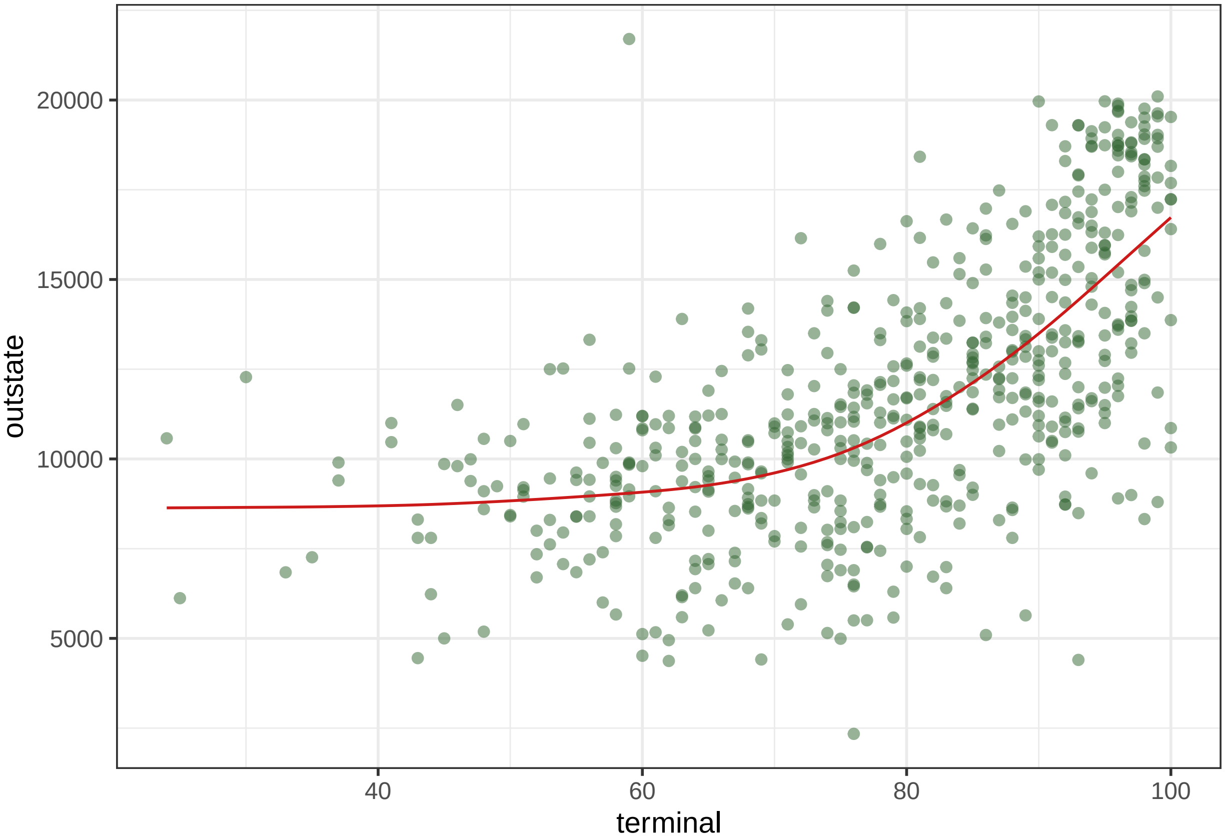


b) Fit smoothing spline models

```
#df
fit.ss <- smooth.spline(college_train$terminal, college_train$outstate)
fit.ss$df
```

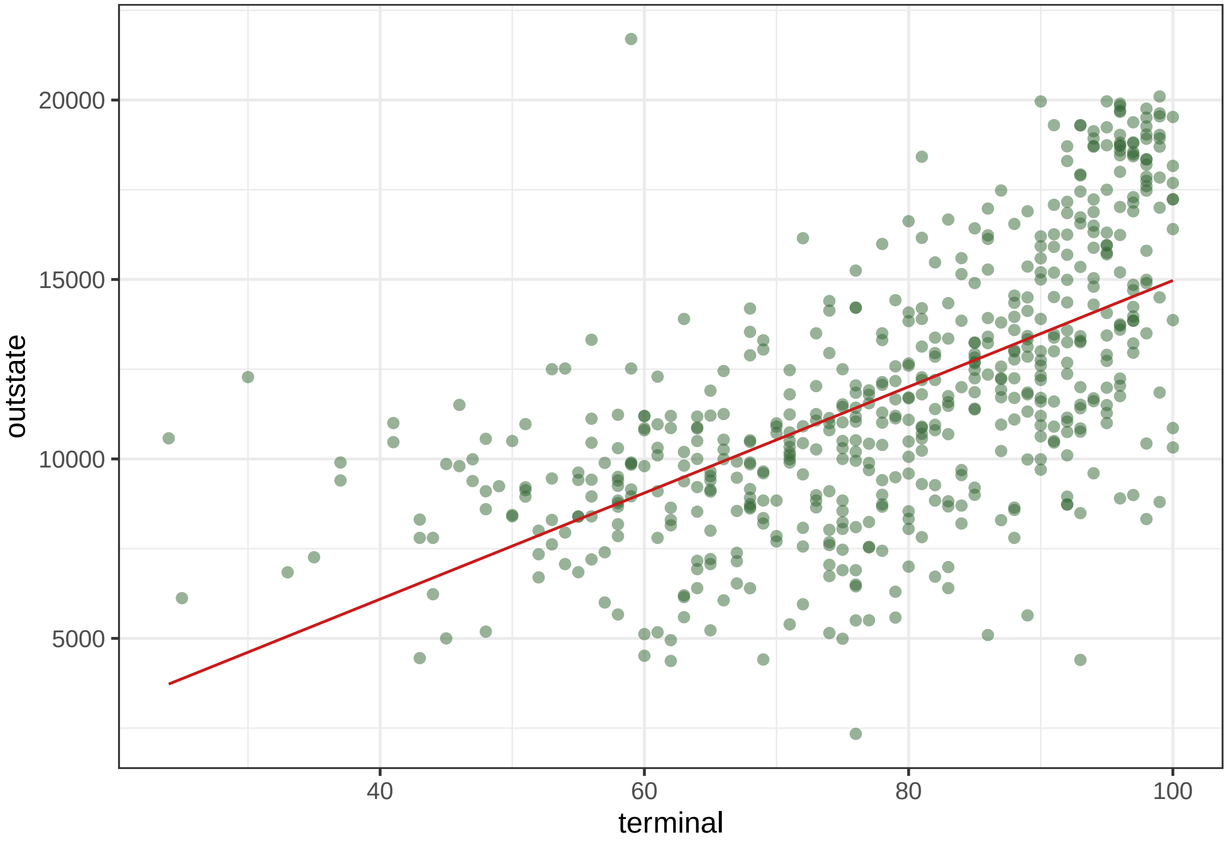
```
## [1] 4.468629
```

```
#plot the resulting fits
terminallims <- range(college_train$terminal)
terminal.grid <- seq(from = terminallims[1],to = terminallims[2])

pred.ss = predict(fit.ss, x = terminal.grid)
pred.ss.df = data.frame(pred = pred.ss$y, terminal = terminal.grid)

p = ggplot(data = college_train, aes(x = terminal, y = outstate)) +
geom_point(color = rgb(.2, .4, .2, .5))

p +
geom_line(aes(x = terminal, y = pred), data = pred.ss.df, color = rgb(.8, .1, .1, 1)) + theme_bw()
```

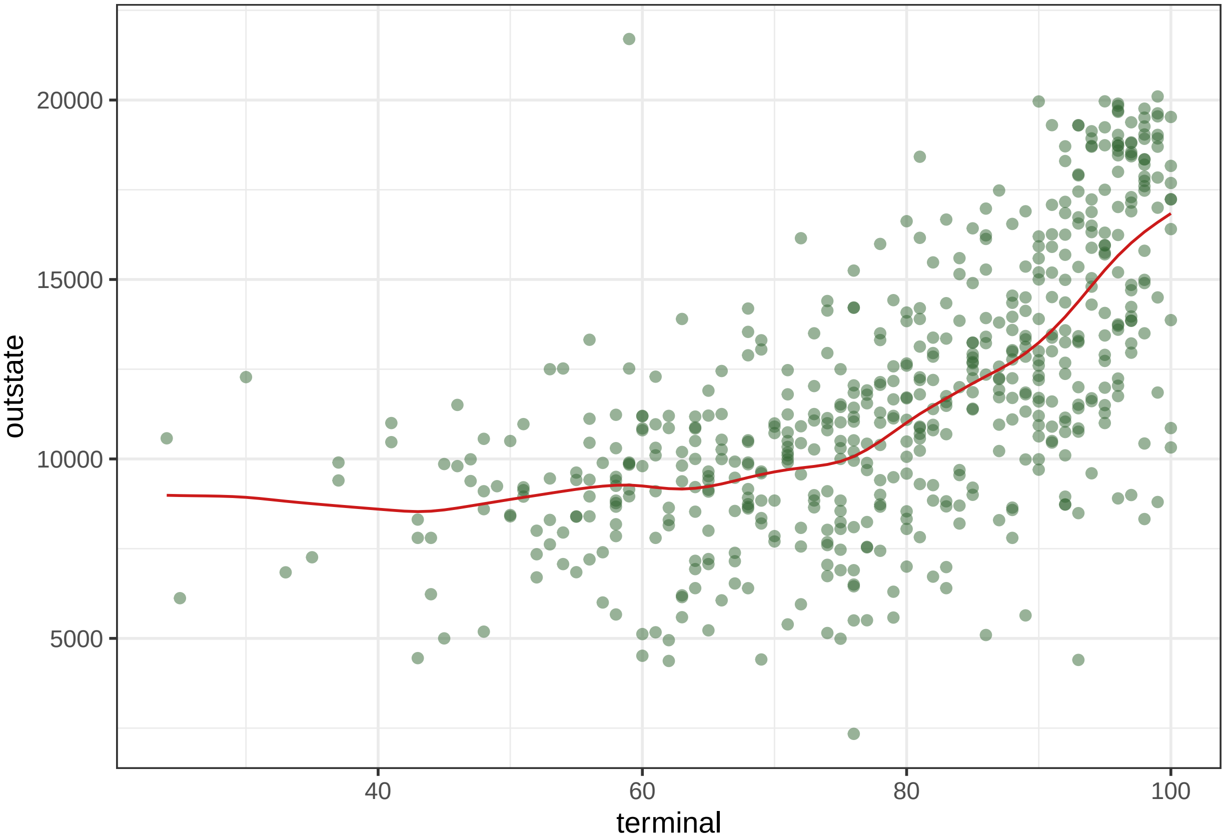


Choose my own degree of freedom:

```
#2 degree of freedom
fit.ss <- smooth.spline(college_train$terminal, college_train$outstate, df=2)
df_2 = fit.ss$df

#plot the resulting fits
terminallims <- range(college_train$terminal)
terminal.grid <- seq(from = terminallims[1],to = terminallims[2])

pred.ss = predict(fit.ss, x = terminal.grid)
pred.ss.df = data.frame(pred = pred.ss$y, terminal = terminal.grid)

p = ggplot(data = college_train, aes(x = terminal, y = outstate)) +
geom_point(color = rgb(.2, .4, .2, .5))

p +
geom_line(aes(x = terminal, y = pred), data = pred.ss.df, color = rgb(.8, .1, .1, 1)) + theme_bw()
```



```
#10 degree of freedom
fit.ss <- smooth.spline(college_train$terminal, college_train$outstate, df=10)
df_10 = fit.ss$df

#plot the resulting fits
terminallims <- range(college_train$terminal)
terminal.grid <- seq(from = terminallims[1],to = terminallims[2])

pred.ss = predict(fit.ss, x = terminal.grid)
pred.ss.df = data.frame(pred = pred.ss$y, terminal = terminal.grid)

p = ggplot(data = college_train, aes(x = terminal, y = outstate)) +
geom_point(color = rgb(.2, .4, .2, .5))

p +
geom_line(aes(x = terminal, y = pred), data = pred.ss.df, color = rgb(.8, .1, .1, 1)) + theme_bw()
```



The degree of freedom obtained by generalized cross validation is 4.468629. From the first plot, we can see that there is a non linear relationship between terminal and outstate. When we only use terminal as a predictor and the degree of freedom obtained by generalized cv to fit the data, the fitted curve is very smooth. The smoothing spline, which is the red line, shows the prediction of the smoothing spline fits the data.

When picking df = 2 and df = 10, we can see that df = 2 shows a linear line, and df = 10 shows a wiggly line. We can conclude that larger values make the line much more wiggly, while smaller degrees of freedom are more linear.

c) Fit a generalized additive model (GAM) using all the predictors

```
ctrl1 <- trainControl(method = "cv", number = 5)
set.seed(7)
gam.fit <- train(x, y,
method = "gam",
tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
trControl = ctrl1)
gam.fit$bestTune
```

```
##   select method
## 1  FALSE GCV.Cp
```

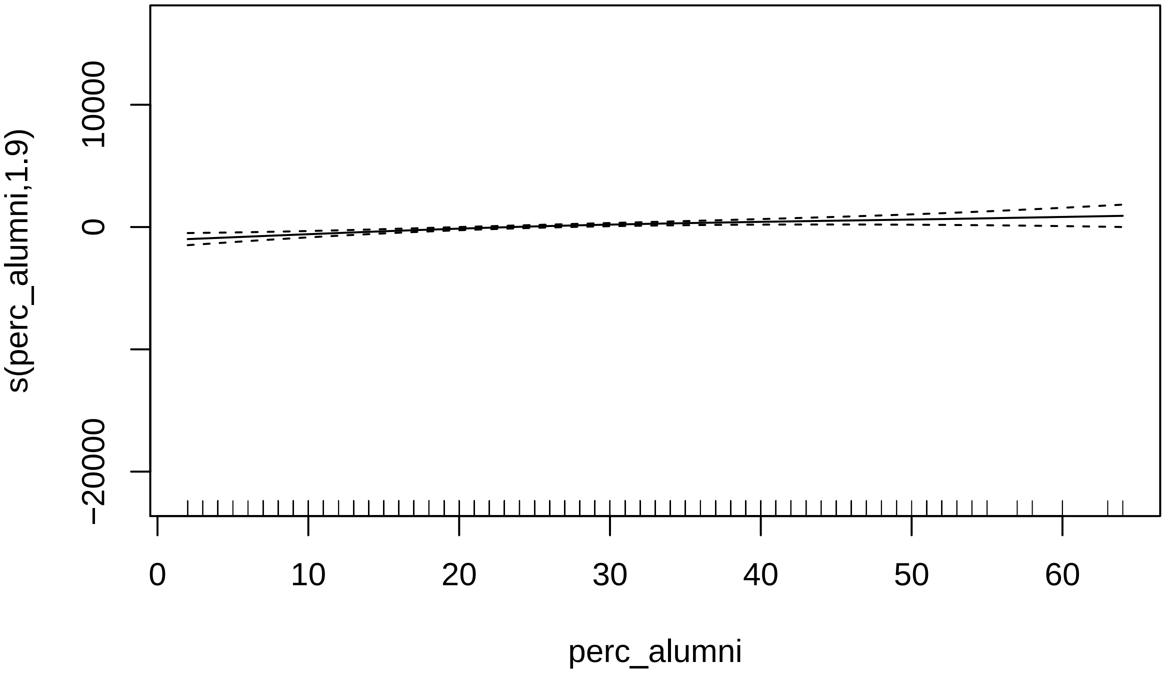
```r
gam.fit$results
```

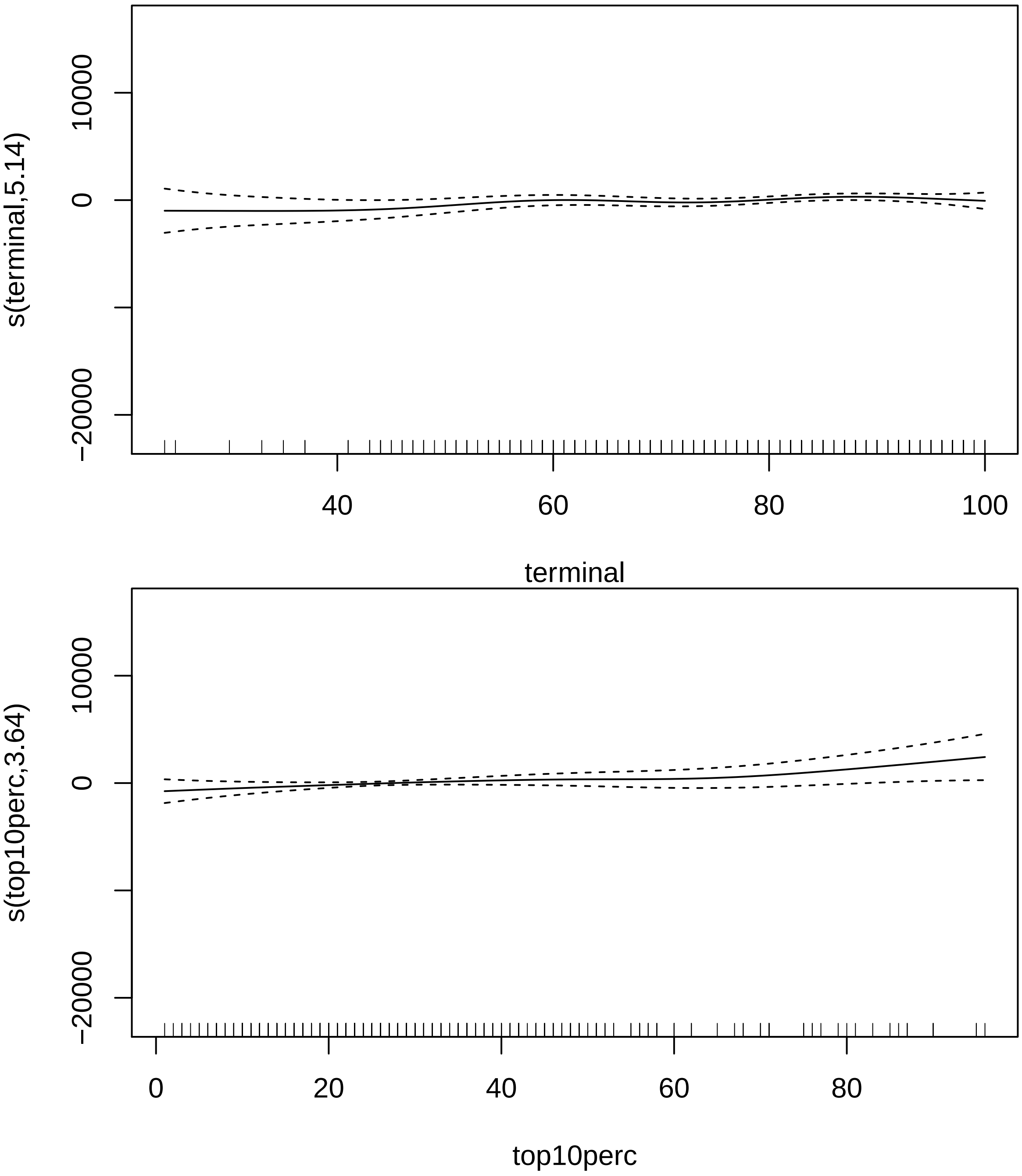
```
##   method select     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1 GCV.Cp  FALSE 1816.709 0.7654877 1384.619 204.1885 0.04437825 107.4501
## 2 GCV.Cp   TRUE 1905.812 0.7476713 1415.683 301.6884 0.06002304 101.9745
```

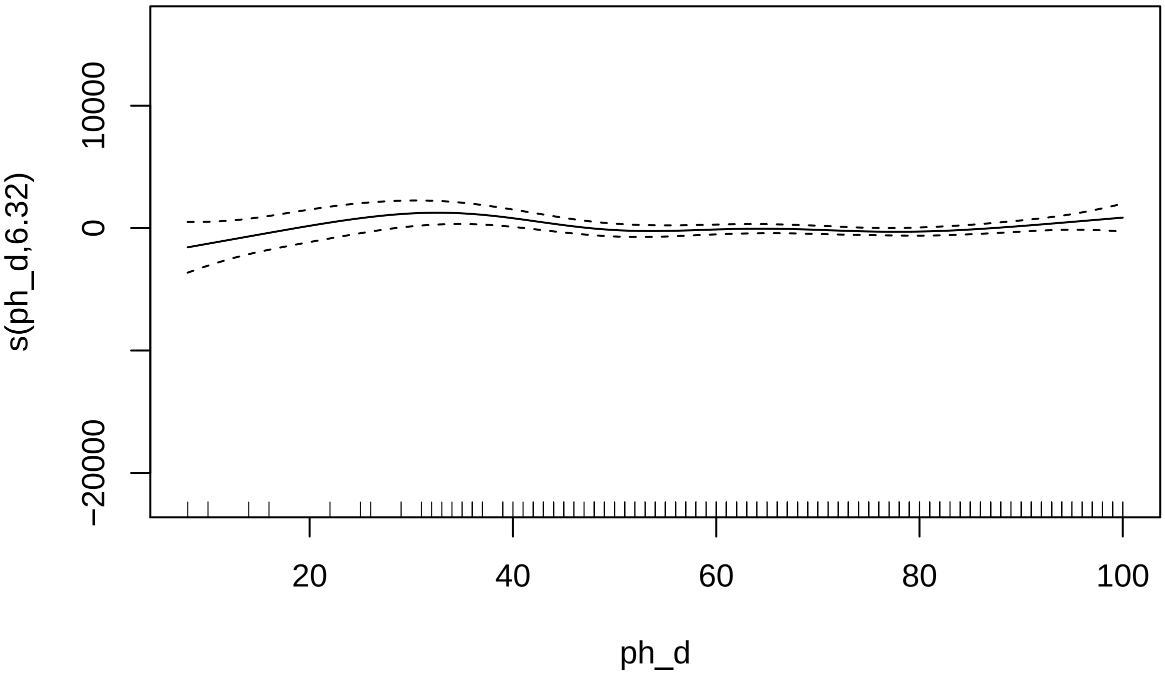
```r
gam.fit$finalModel
```

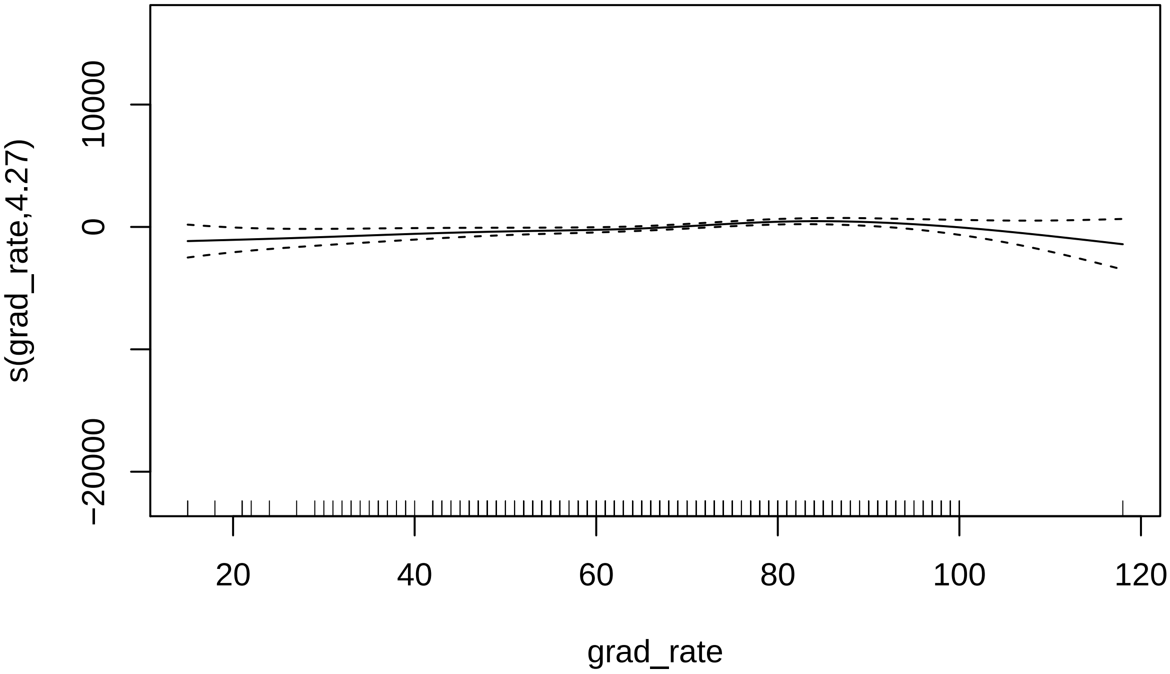
```
## 
## Family: gaussian 
## Link function: identity 
## 
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(top10perc) + s(ph_d) + 
##     s(grad_rate) + s(books) + s(top25perc) + s(s_f_ratio) + s(personal) + 
##     s(p_undergrad) + s(enroll) + s(room_board) + s(accept) + 
##     s(f_undergrad) + s(apps) + s(expend)
## 
## Estimated degrees of freedom:
## 1.90 5.14 3.64 6.32 4.27 2.35 1.00 
## 4.33 1.00 1.00 1.00 2.13 3.58 6.28 
## 4.59 6.45  total = 55.98 
## 
## GCV score: 2761951
```

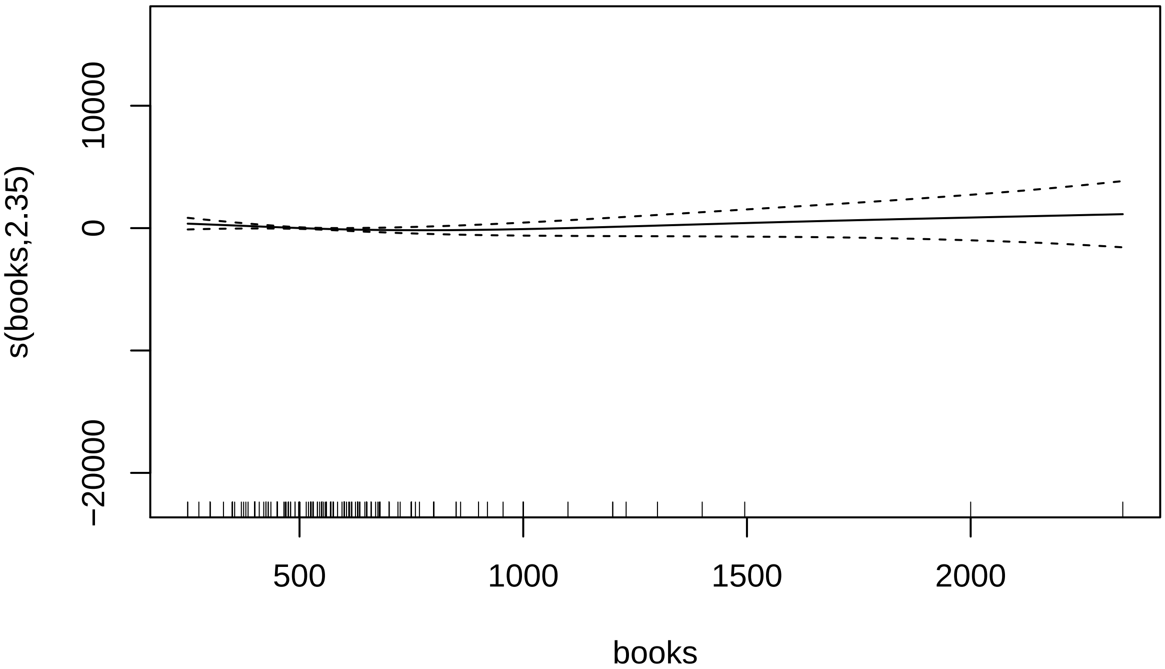
```r
plot(gam.fit$finalModel)
```

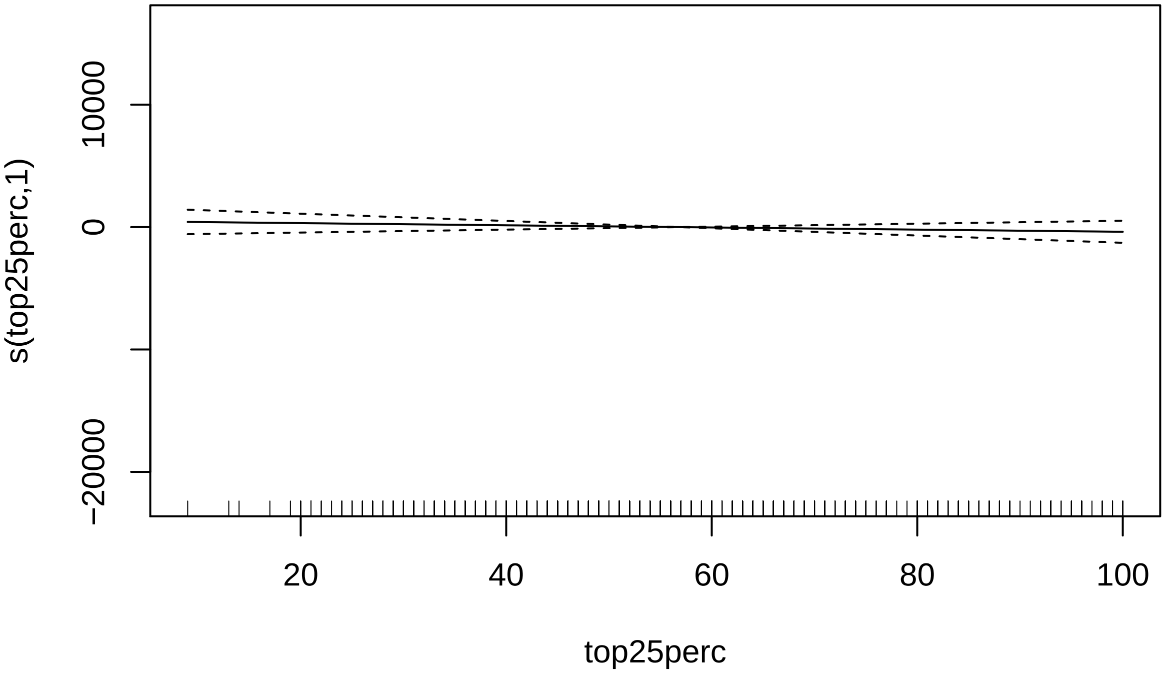


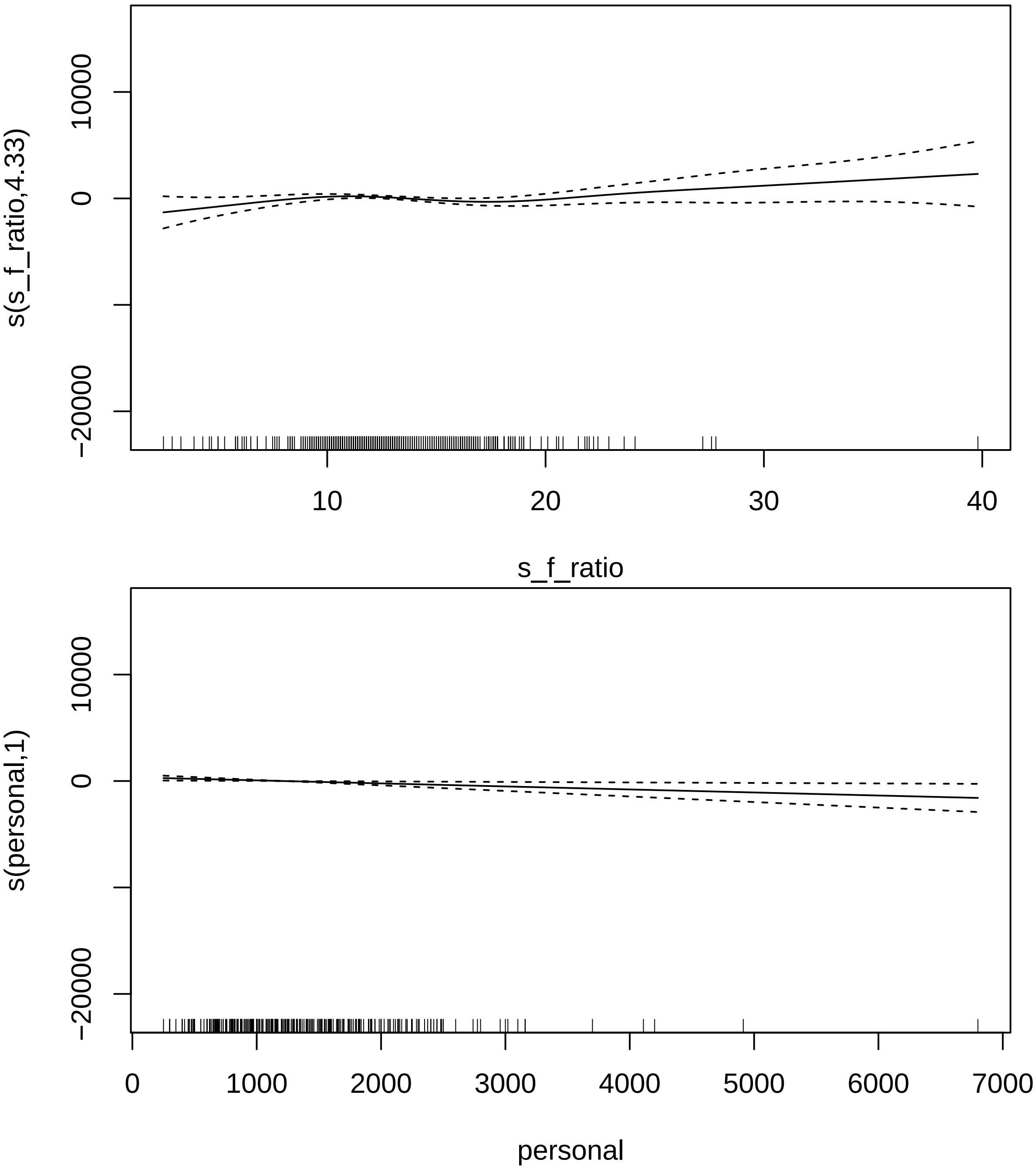

s(terminal,5.14)
10000
0
−20000
40
60
80
100
terminal
s(top10perc,3.64)
10000
0
−20000
0
20
40
60
80
top10perc

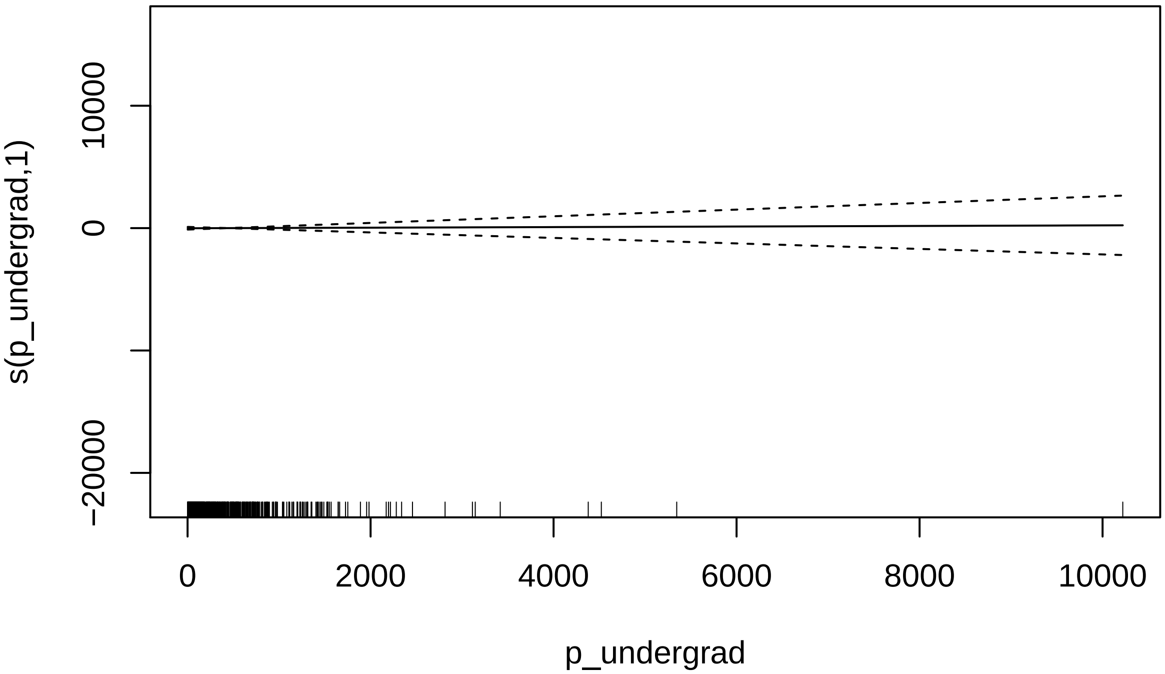

s(ph_d,6.32)
10000
0
−20000
20
40
60
80
100
ph_d

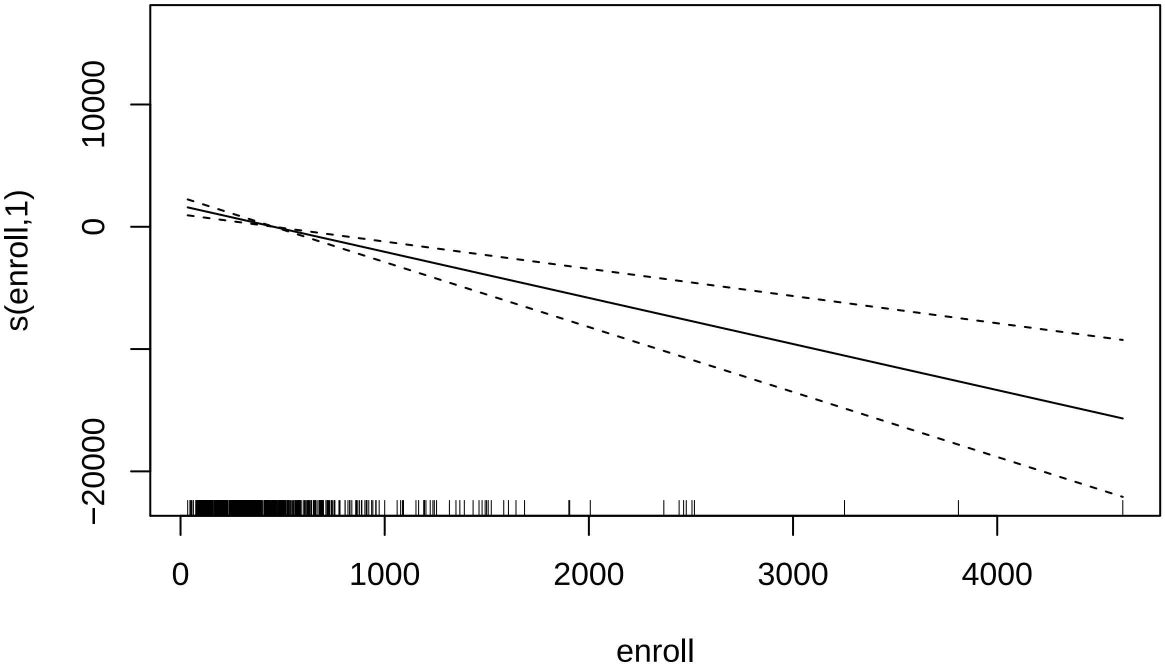

s(grad_rate,4.27)
10000
0
−20000
20
40
60
80
100
120
grad_rate

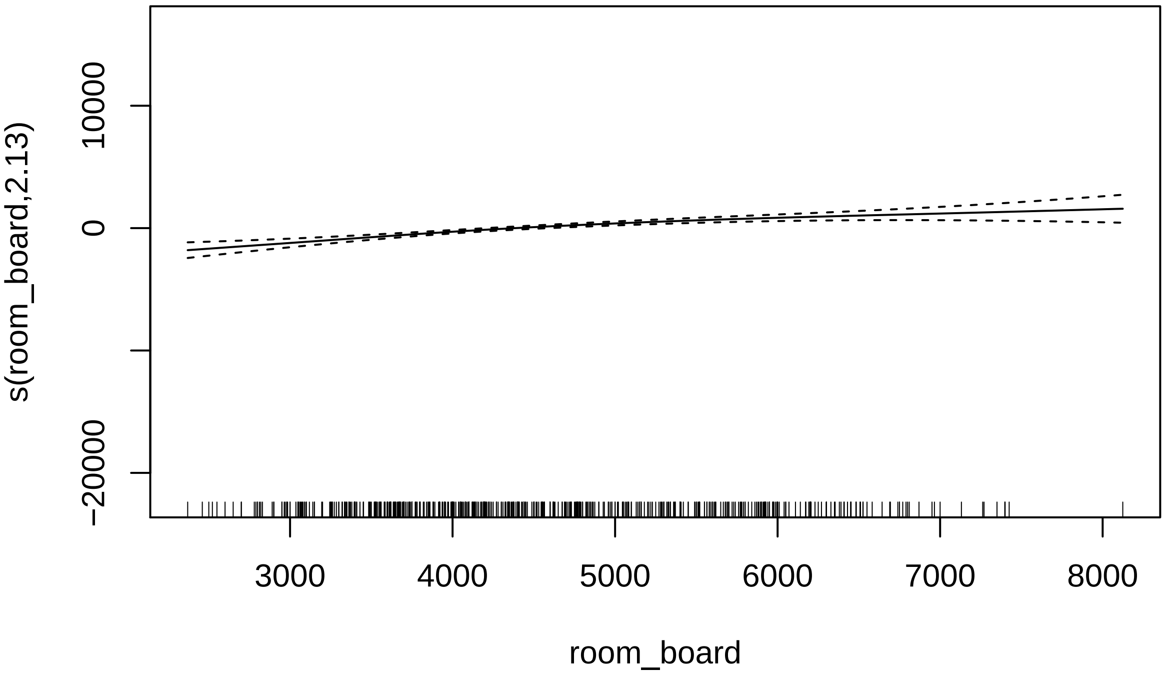

s(books,2.35)
10000
0
−20000
500
1000
1500
2000
books

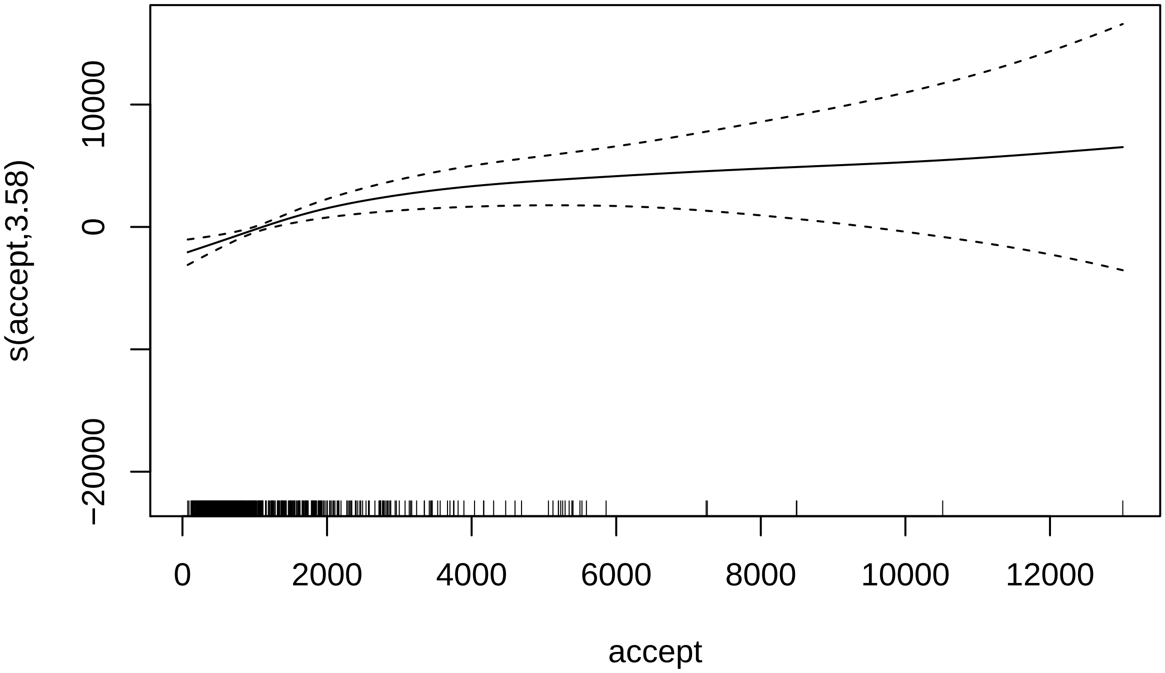

s(top25perc,1)
10000
0
−20000
20
40
60
80
100
top25perc

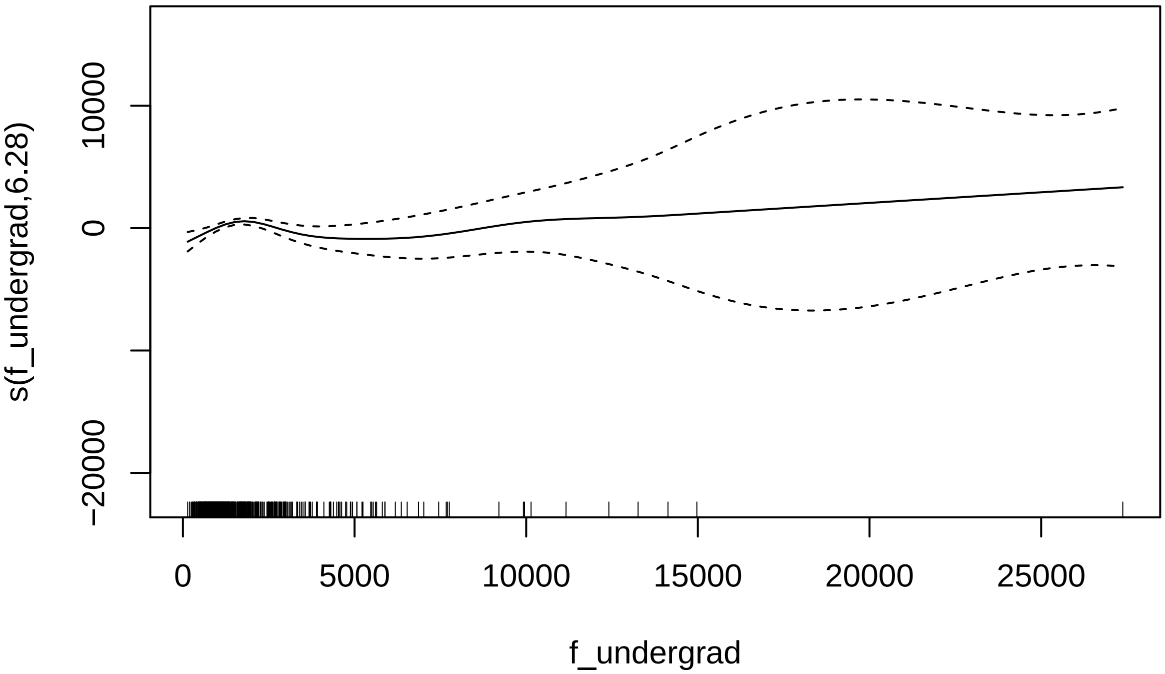

s(s_f_ratio,4.33)
10000
0
−20000
10
20
30
40
s_f_ratio
s(personal,1)
10000
0
−20000
0
1000
2000
3000
4000
5000
6000
7000
personal

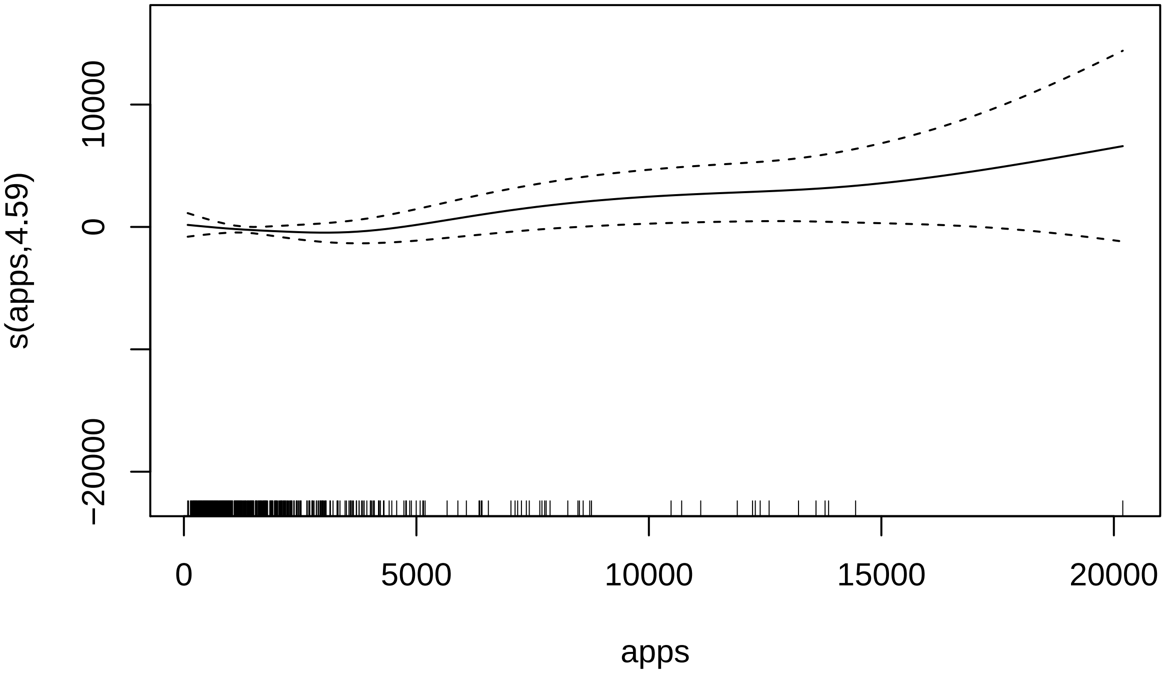

s(p_undergrad,1)
10000
0
−20000
0
2000
4000
6000
8000
10000
p_undergrad

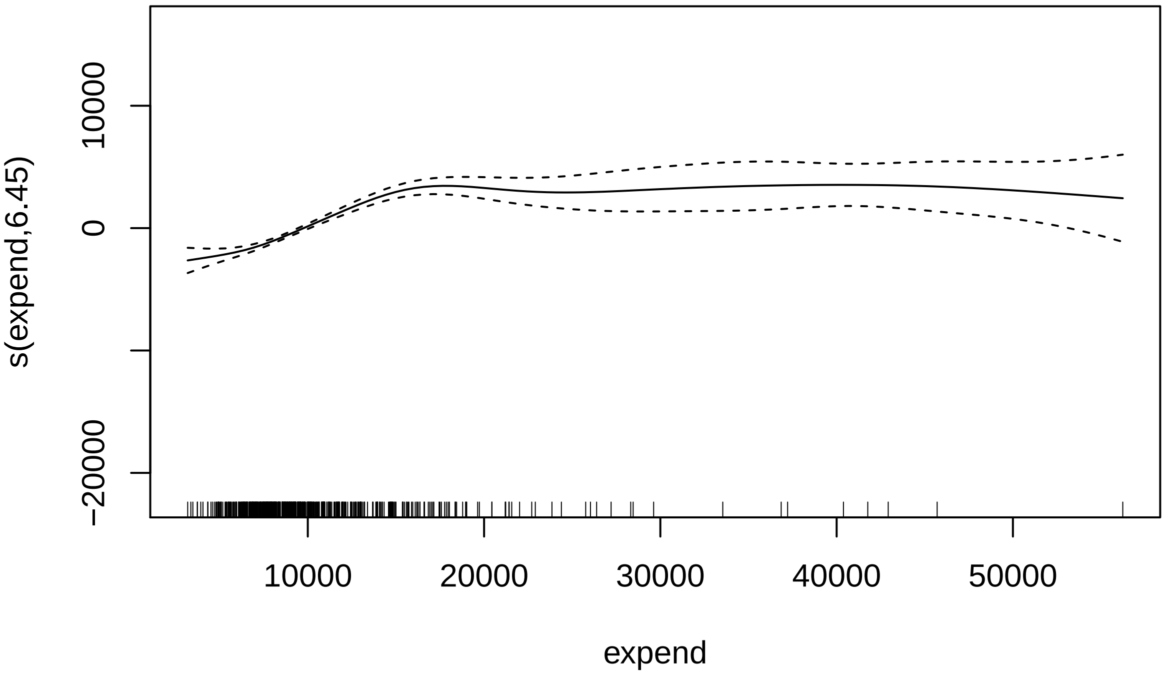

s(enroll,1)
10000
0
−20000
0
1000
2000
3000
4000
enroll


s(room_board,2.13)
10000
0
−20000
3000
4000
5000
6000
7000
8000
room_board


s(accept,3.58)
10000
0
−20000
0
2000
4000
6000
8000
10000
12000
accept


s(f_undergrad,6.28)
10000
0
−20000
0
5000
10000
15000
20000
25000
f_undergrad

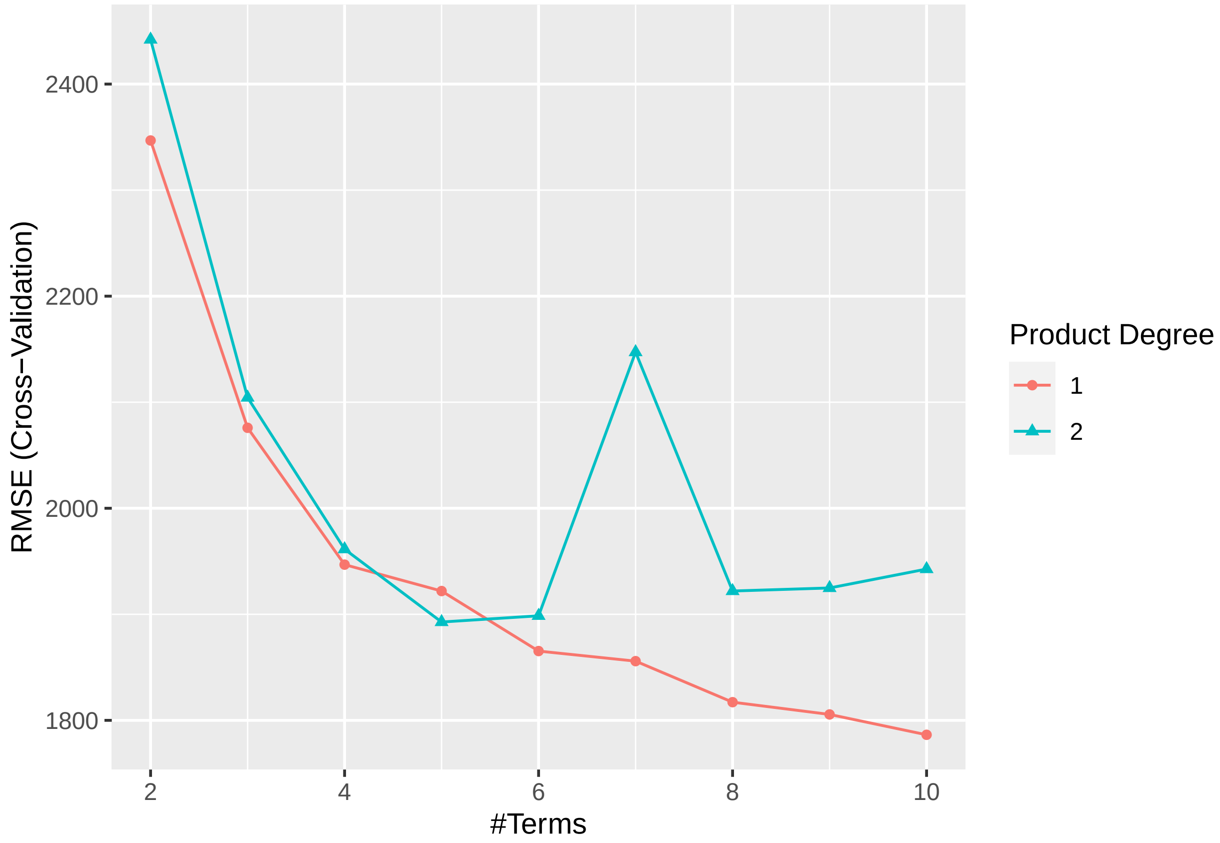

s(apps,4.59)
10000
0
−20000
0
5000
10000
15000
20000
apps



From the results by caret, we can see that the output of bestTune showed that selecting FALSE is better than selecting TRUE. Comparing the results in caret, RMSE of selecting FALSE is smaller than that of selecting TRUE. From the final model, we can see that it added smooth function to every variable. Both GCV and df score are large. The result of caret also indicates that there are some potential tensor interaction between predictors.

d) Train a multivariate adaptive regression spline (MARS) model using all the predictors

```
ctrl1 <- trainControl(method = "cv", number = 5)

mars_grid <- expand.grid(degree = 1:2,
                         nprune = 2:10)
set.seed(1)
mars.fit <- train(x, y,
                 method = "earth",
                 tuneGrid = mars_grid,
                 trControl = ctrl1)

ggplot(mars.fit)
```



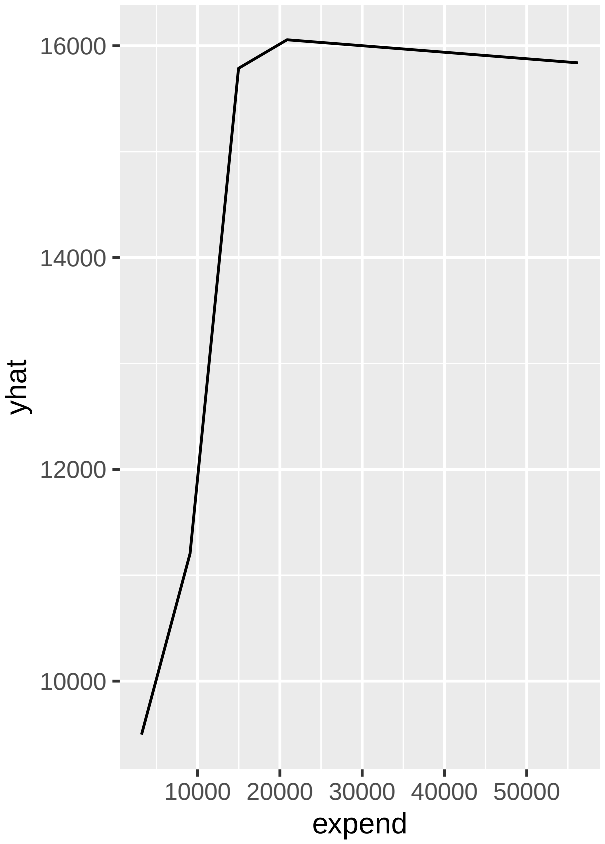
```
mars.fit$bestTune
```

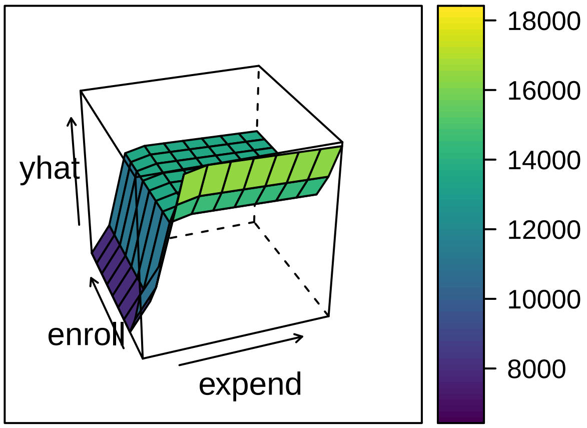
```
##   nprune degree
## 9     10      1
```

```
coef(mars.fit$finalModel)
```

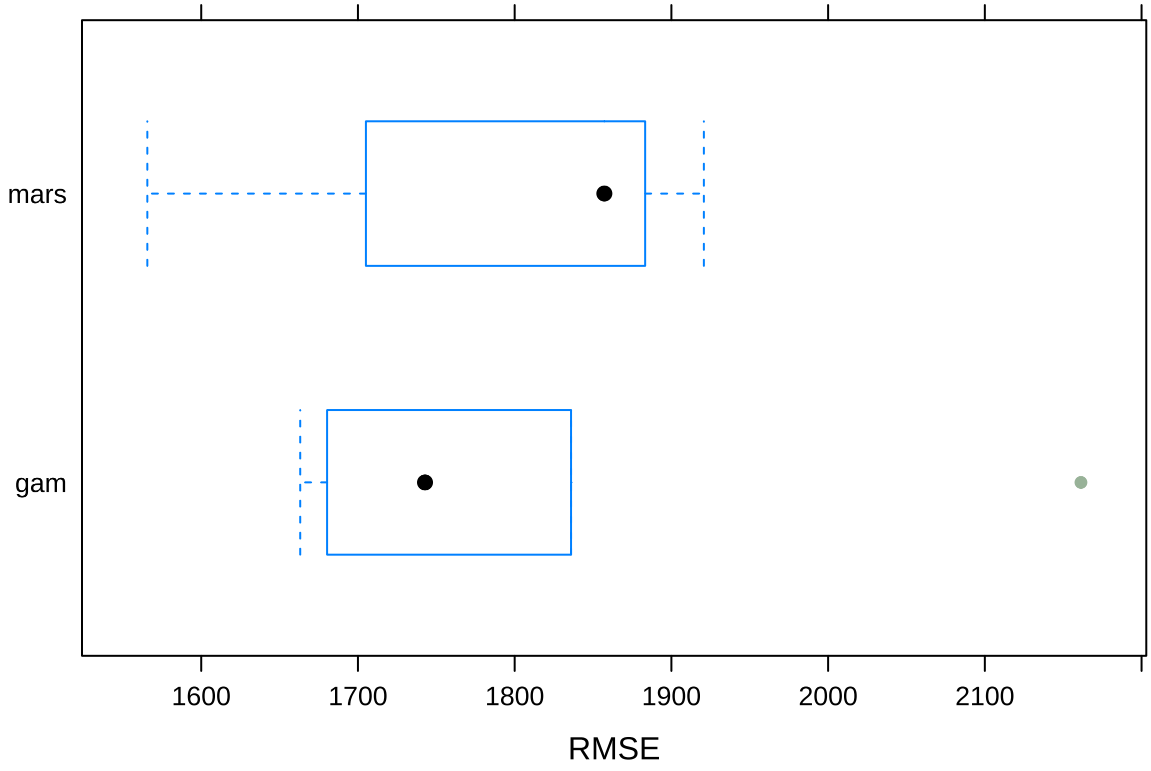
```
##         (Intercept)     h(expend-15365)  h(4450-room_board) h(f_undergrad-1355)
##       10856.8275542          -0.7836173          -1.4272043          -0.3818847
## h(1355-f_undergrad)   h(22-perc_alumni)        h(apps-3712)       h(913-enroll)
##          -1.6799143        -105.5570689           0.4334737           4.5019587
##      h(2193-accept)      h(expend-6881)
##          -1.9769988           0.7774546
```

```
#partial dependence plot
p1 = pdp::partial(mars.fit, pred.var = c("expend"), grid.resolution = 10) %>% autoplot()

p2 = pdp::partial(mars.fit, pred.var =c("expend","enroll"), grid.resolution = 10) %>% plotPartial(levelp

grid.arrange(p1, p2, ncol = 2)
```





```
bwplot(resamples(list(mars = mars.fit,
gam = gam.fit)), metric = "RMSE")
```



The final model using MARS is:

f(x) = 10856.83 - 0.78h(expend-15365) - 1.43h(4450-room_board) - 0.38h(f_undergrad-1355) - 1.68h(1355-f_undergrad) - 105.56h(22-perc_alumni) + 0.43h(apps-3712) + 4.50h(913-enroll) - 1.97h(2193-accept) + 0.78h(expend-6881)

From the boxplot, we can see that GAM has smaller RMSE than MARS.