HW2_sz2800

Stephanie Zhen

3/19/2020

Loading libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

```
## -- Conflicts -------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(splines)
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
## This is mgcv 1.8-31. For overview type 'help("mgcv-package")'.
```

```r
library(earth)
```

```
## Warning: package 'earth' was built under R version 3.6.3
```

```
## Loading required package: Formula
```

```
## Loading required package: plotmo
```

```
## Warning: package 'plotmo' was built under R version 3.6.3
```

```
## Loading required package: plotrix
```

```
## Loading required package: TeachingDemos
```

```
## Warning: package 'TeachingDemos' was built under R version 3.6.3
```

```r
library(pdp)
```

```
## Warning: package 'pdp' was built under R version 3.6.3
```

```
##
## Attaching package: 'pdp'
```

```
## The following object is masked from 'package:purrr':
##
##     partial
```

Loading Data and cleaning.

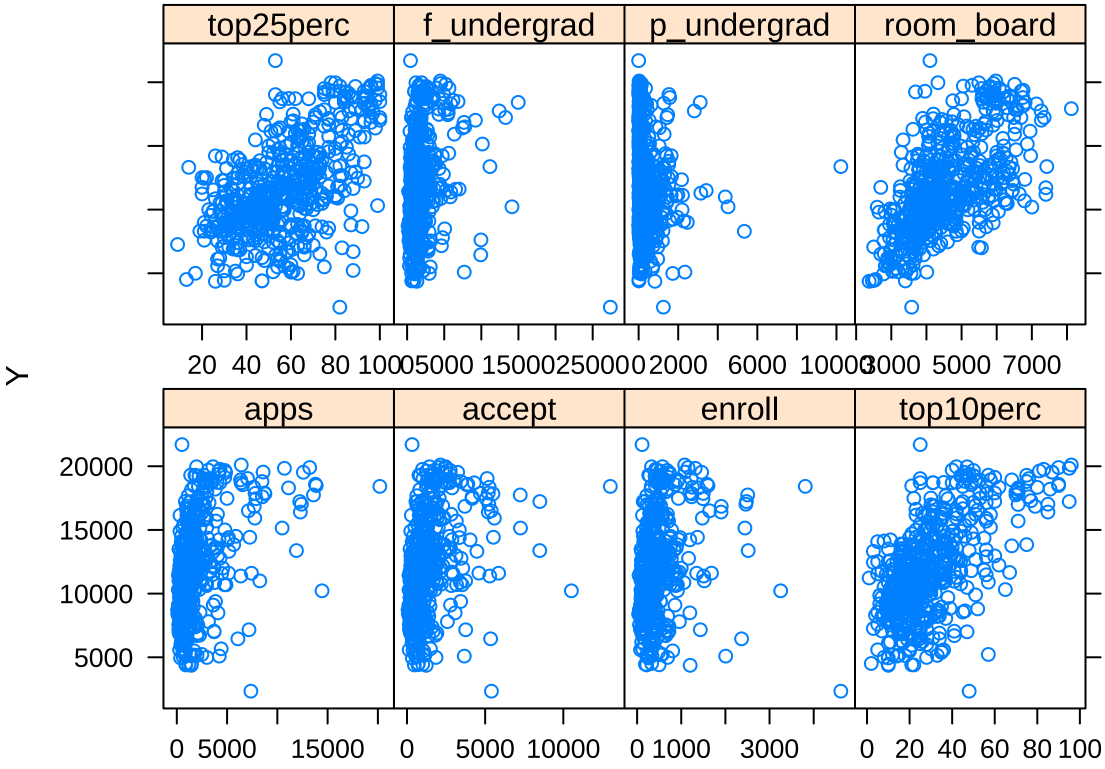
```r
college = read.csv("./College.csv") %>%
  janitor::clean_names()

#Removing the 125 observation (Columbia) for training purposes. Also removed college names columns.
college_df1 = college[-125,-1]

#New dataset for the observation 125, Columbia for prediction purposes. Also removed college names colu
college_cumc = college[125,-1]

#Looking at the structure of the data.
#str(college_df1)
#summary(college_df1)
```

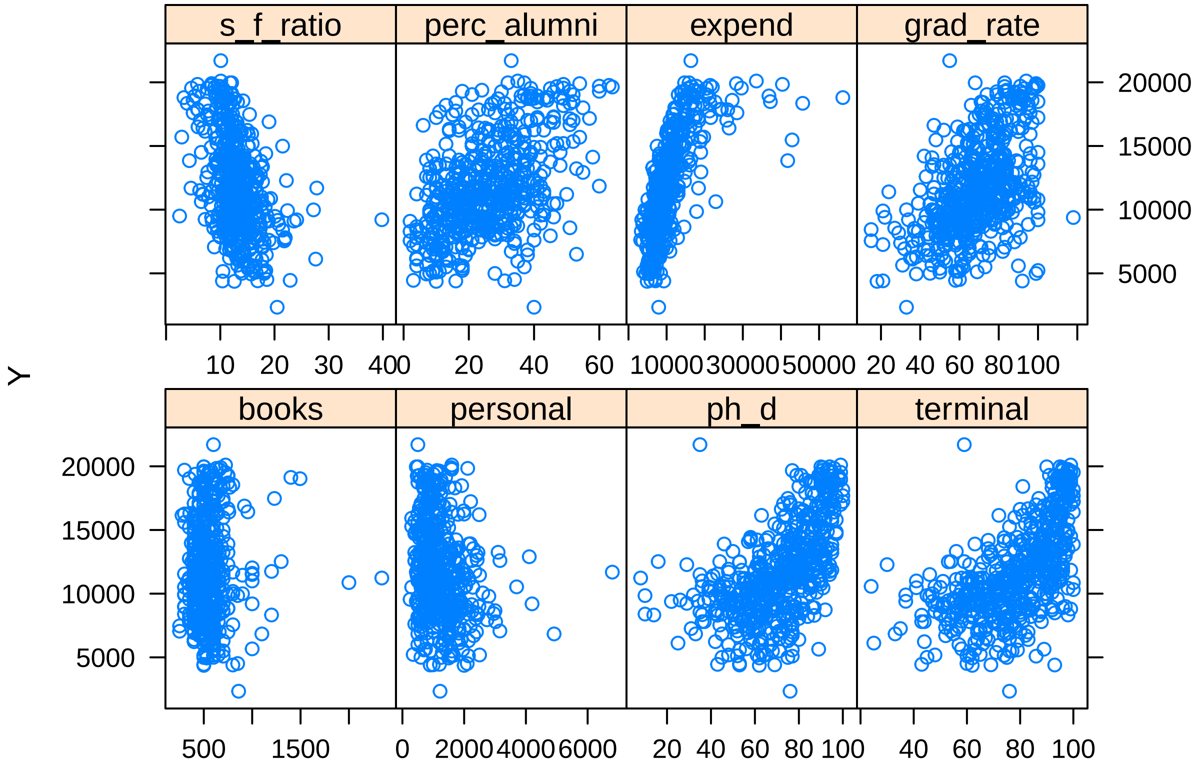
Step 0: Partitioning the outcome variable and the predictors.

```r
#response variable for training
y = college_df1$outstate

#matrix for predictors for training
x = model.matrix(outstate ~ ., data = college_df1)[,-1]
```

Part A. Creating scatterplot of response vs all the predictors.

```r
#Creating scatterplot of response vs predictors.
#There is a total of 16 predictors with 1 response variable, outstate.
featurePlot(x, y, plot = "scatter", labels = c("", "Y"),
            type = c("p"), layout = c(4, 2))
```



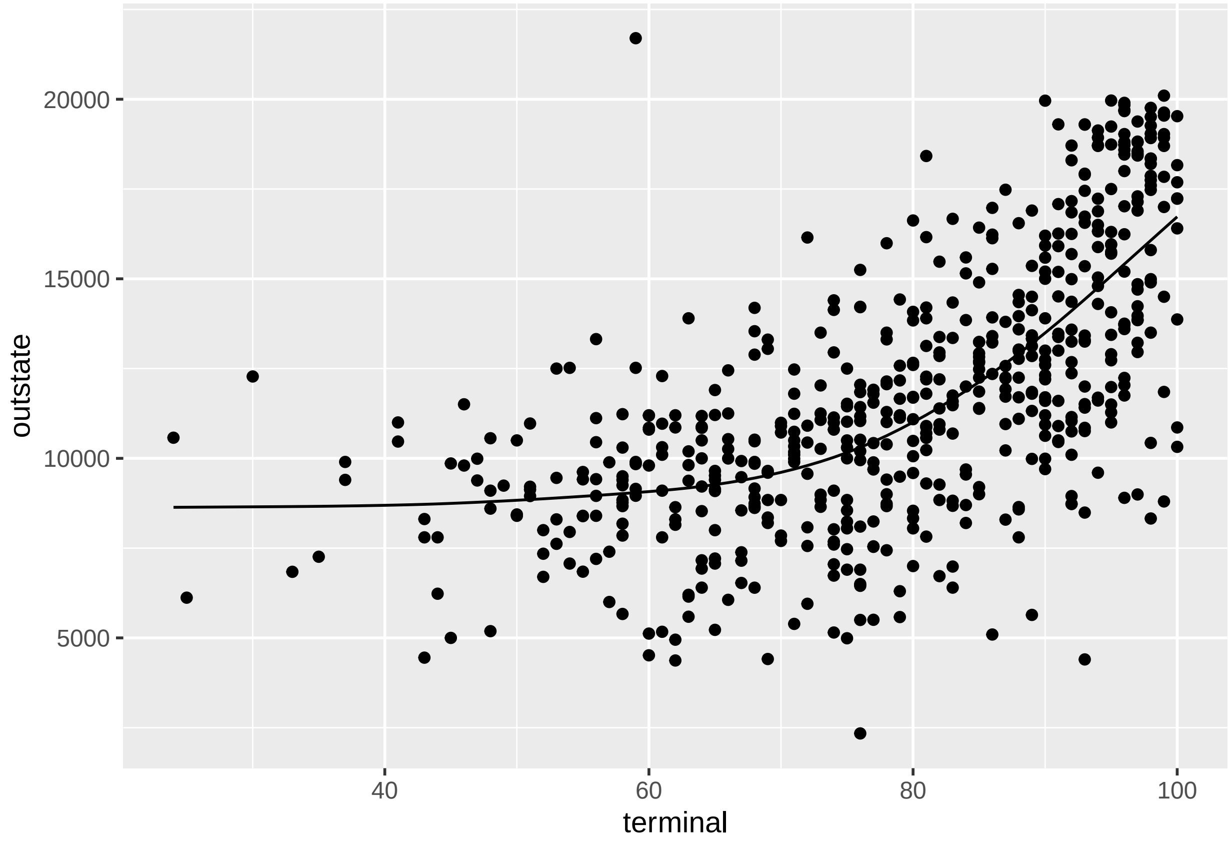


Part B: Smoothing Spline

```
#Using range to predict df for smoothing spline
terminal_range = range(college_df1$terminal)

terminal_grid = seq(from = terminal_range[1],to = terminal_range[2])

#USing GCV to select the degree of freedom (trace of a smoother matrix)
fit_ss = smooth.spline(college_df1$terminal, college_df1$outstate)
fit_ss$df
```

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

```
pred_ss = predict(fit_ss,
                  x = terminal_grid)

pred_ss_df = data.frame(pred = pred_ss$y,
                        terminal = terminal_grid)

#plotting the fit
plot1 = ggplot(data = college_df1,
               aes(x = terminal,
                   y = outstate)) +
        geom_point()
##Adding plot with smooth spline
plot1 + geom_line(aes(x = terminal,
                      y = pred),
                  data = pred_ss_df)
```



Through generalized cross validation the lambda selected was 0.03936 with degree of freedom 4.4686. Given these tuning parameter and the visual of the plot, it seems that the model is very smooth and fits the data.

Part C. GAM

```
#Fitting a generalized additive model (GAM) using all the predictors.

set.seed(123)
control_1 = trainControl(method = "cv", number = 10)
gam_fit = train(x, y,
                method = "gam",
                tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE, FALSE)),
                trControl = control_1)
```

```
## Warning: model fit failed for Fold04: method=GCV.Cp, select= TRUE Error in magic(G$y, G$X, msp, G$S,
##   magic, the gcv/ubre optimizer, failed to converge after 400 iterations.

## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info = trainInfo, :
## There were missing values in resampled performance measures.
```

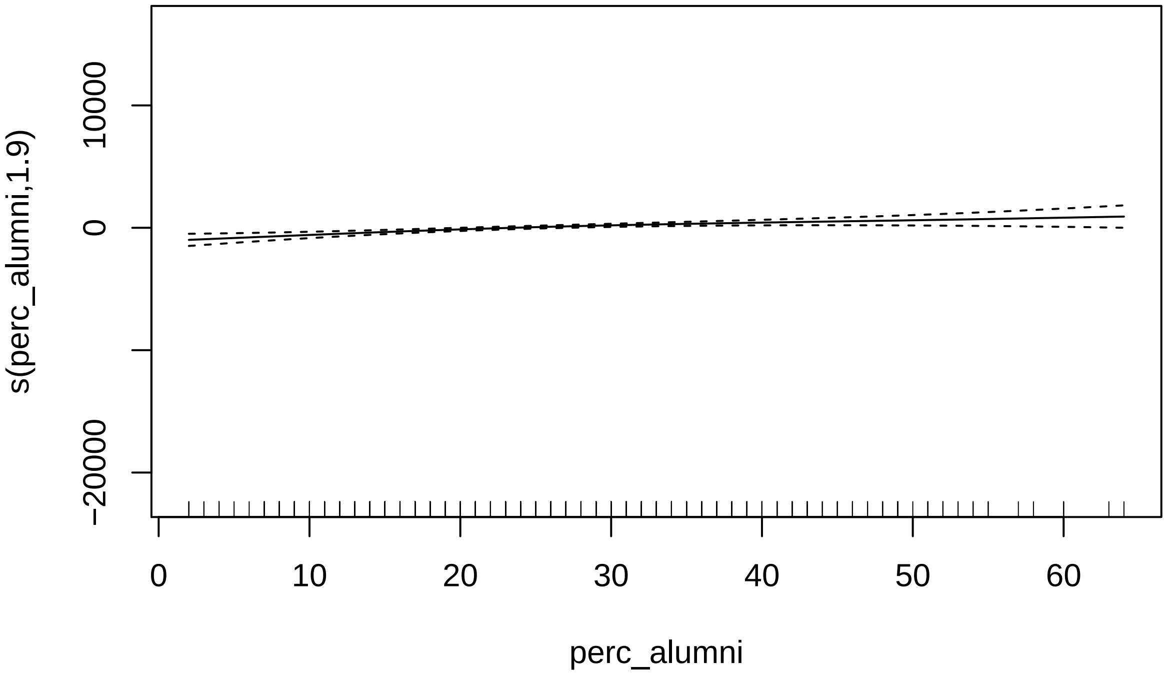
```r
##best tune
gam_fit$bestTune
```

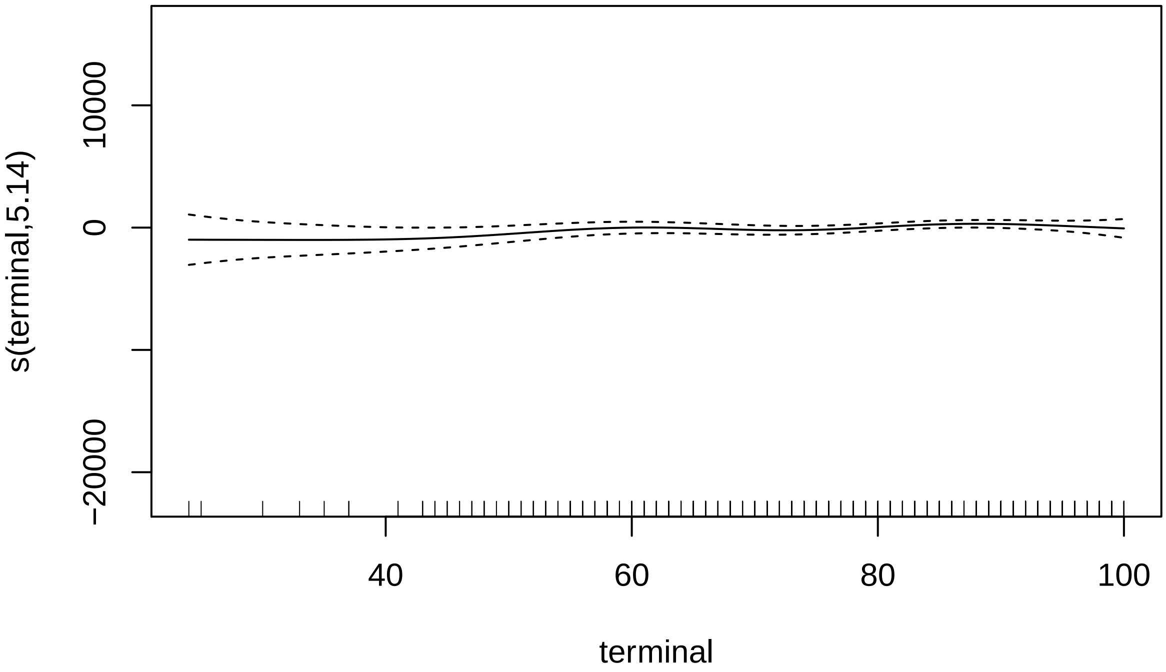
```
##   select method
## 1  FALSE GCV.Cp
```

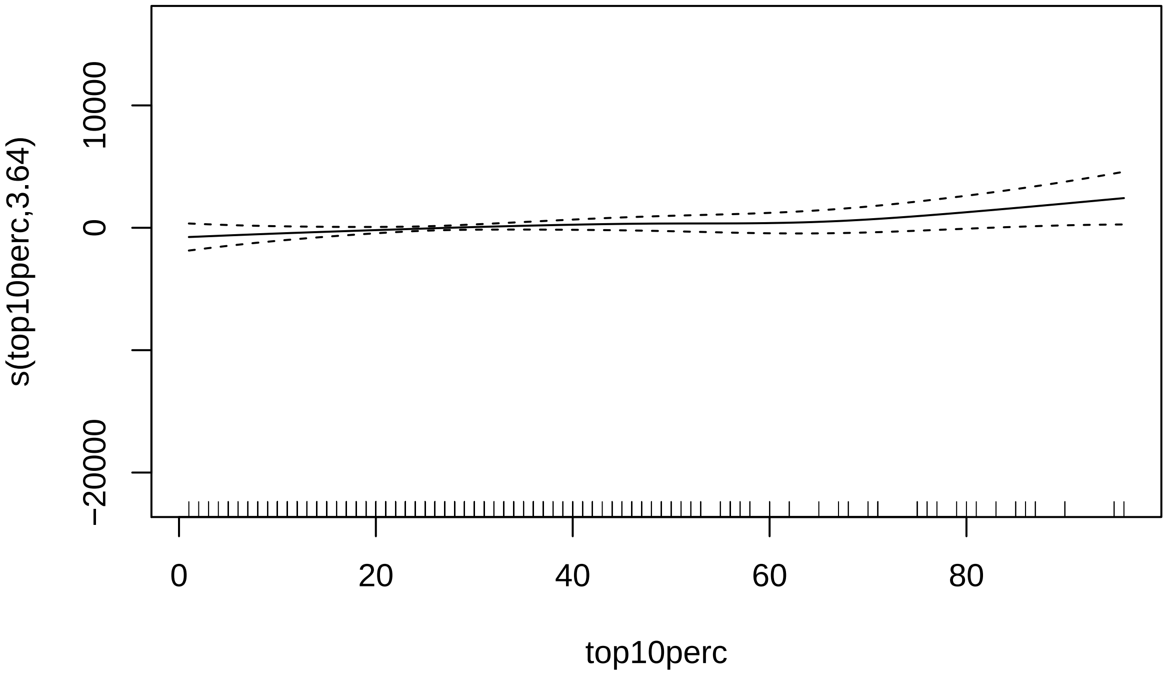
```r
##final model
gam_fit$finalModel
```

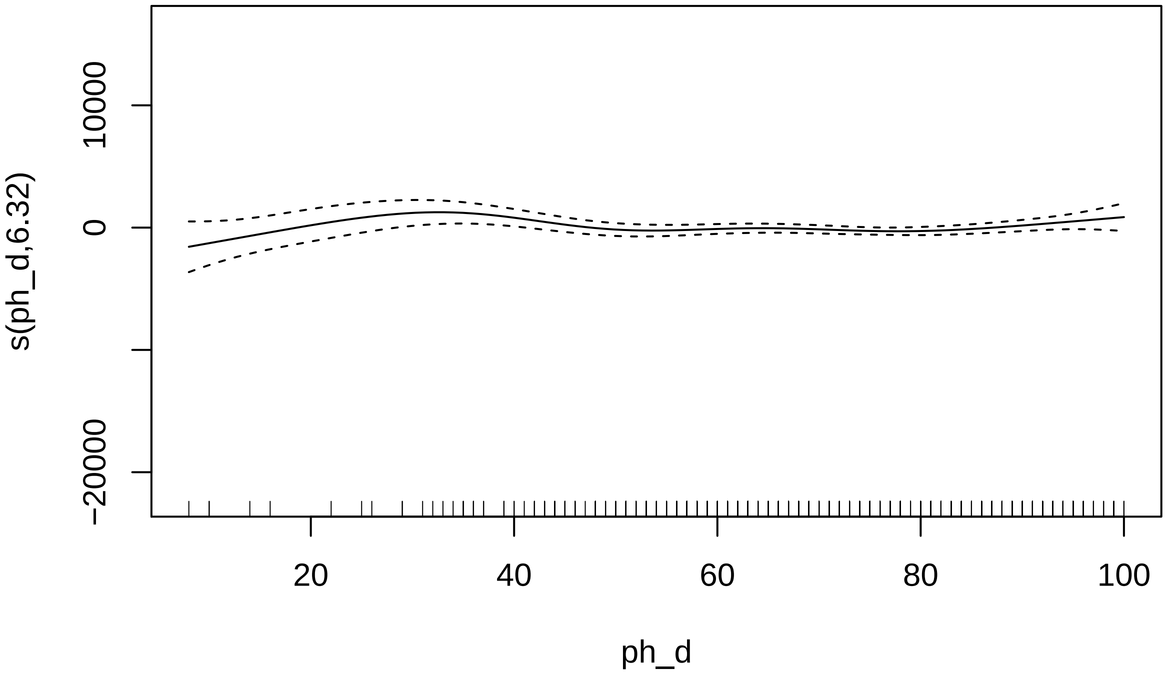
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(top10perc) + s(ph_d) +
##     s(grad_rate) + s(books) + s(top25perc) + s(s_f_ratio) + s(personal) +
##     s(p_undergrad) + s(enroll) + s(room_board) + s(accept) +
##     s(f_undergrad) + 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
```

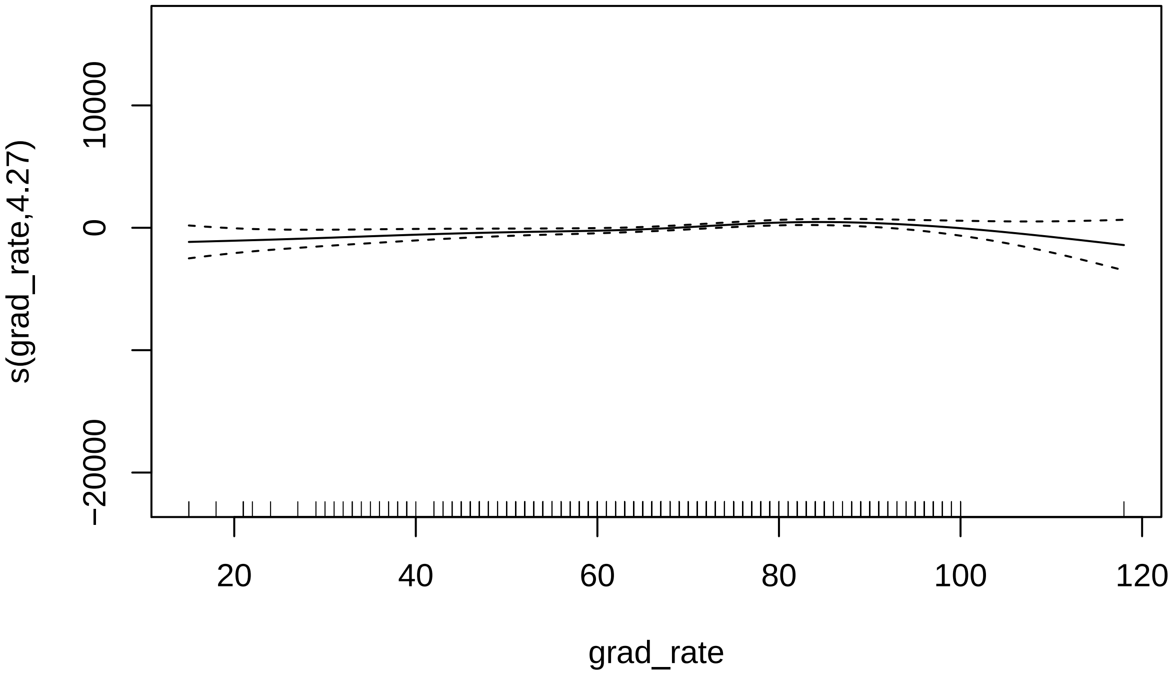
```r
plot(gam_fit$finalModel)
```

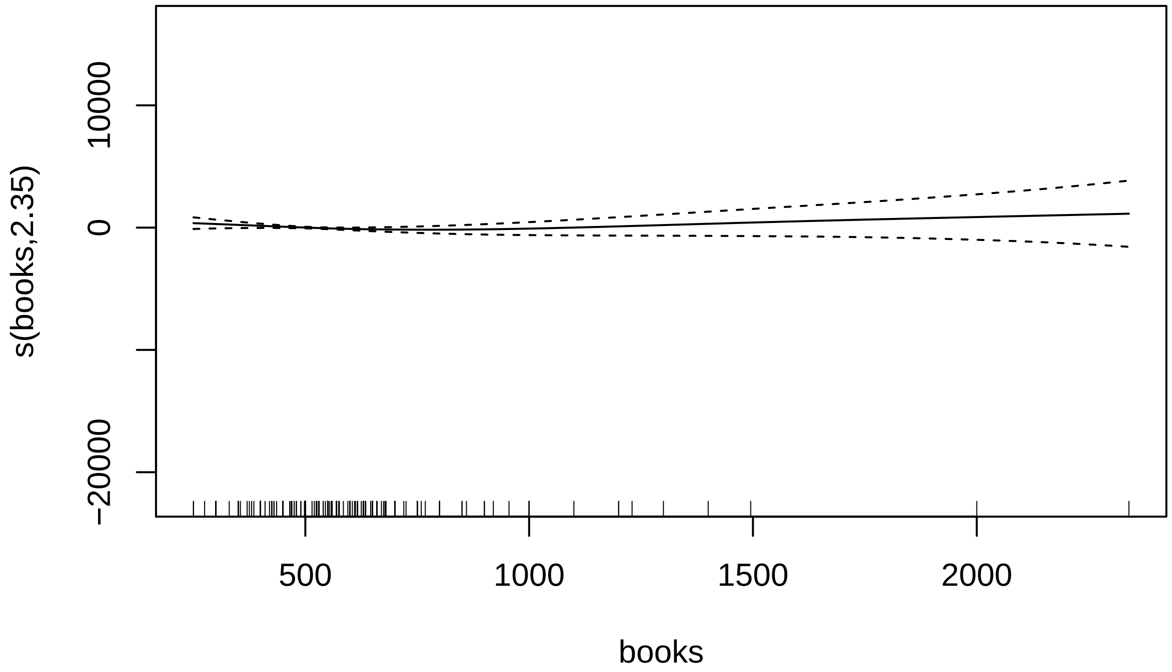

s(perc_alumni,1.9)
10000
0
−20000
0
10
20
30
40
50
60
perc_alumni

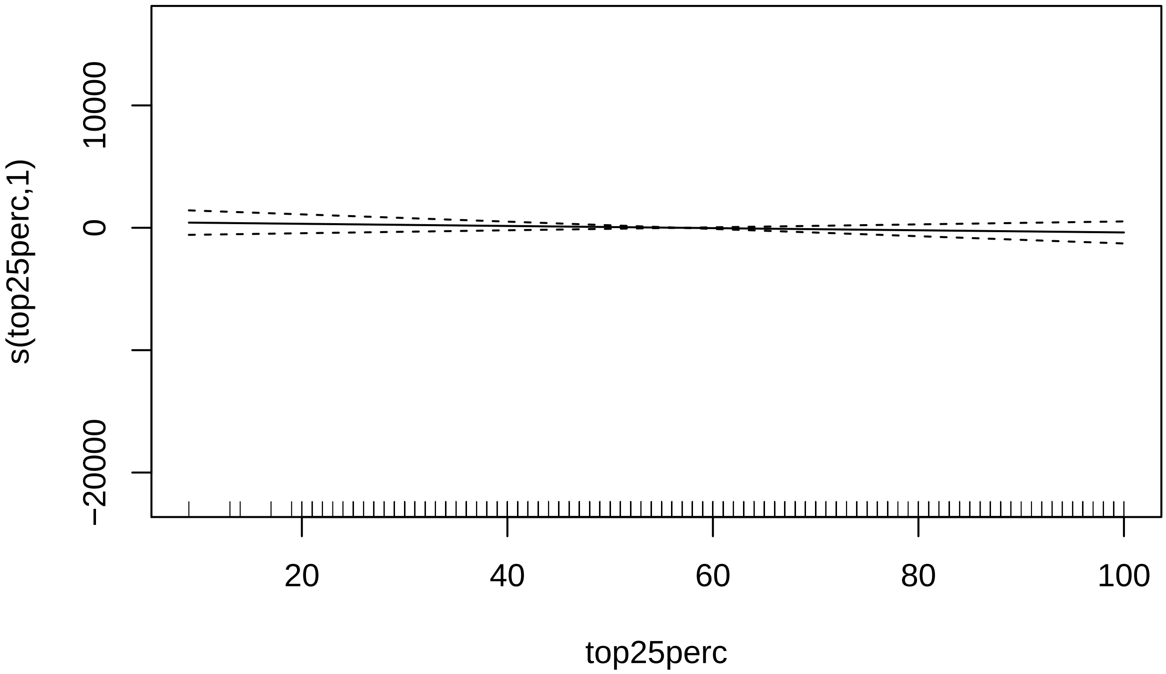

s(terminal,5.14)
10000
0
−20000
40
60
80
100
terminal

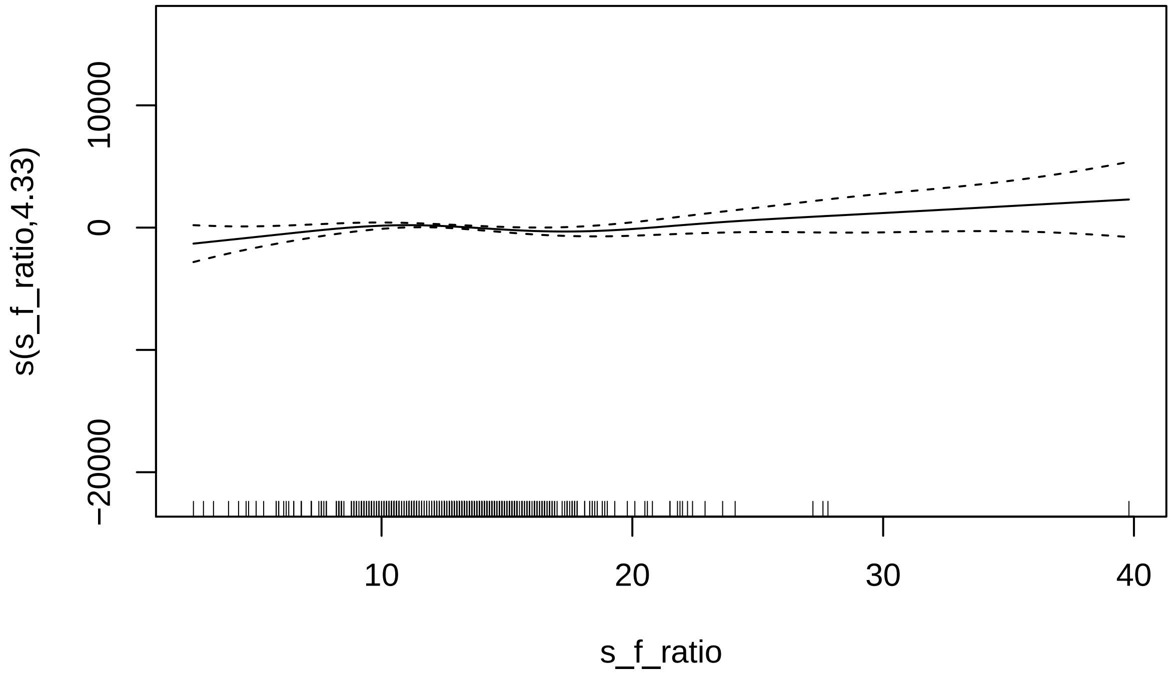

s(top10perc,3.64)
10000
0
−20000
0
20
40
60
80
top10perc

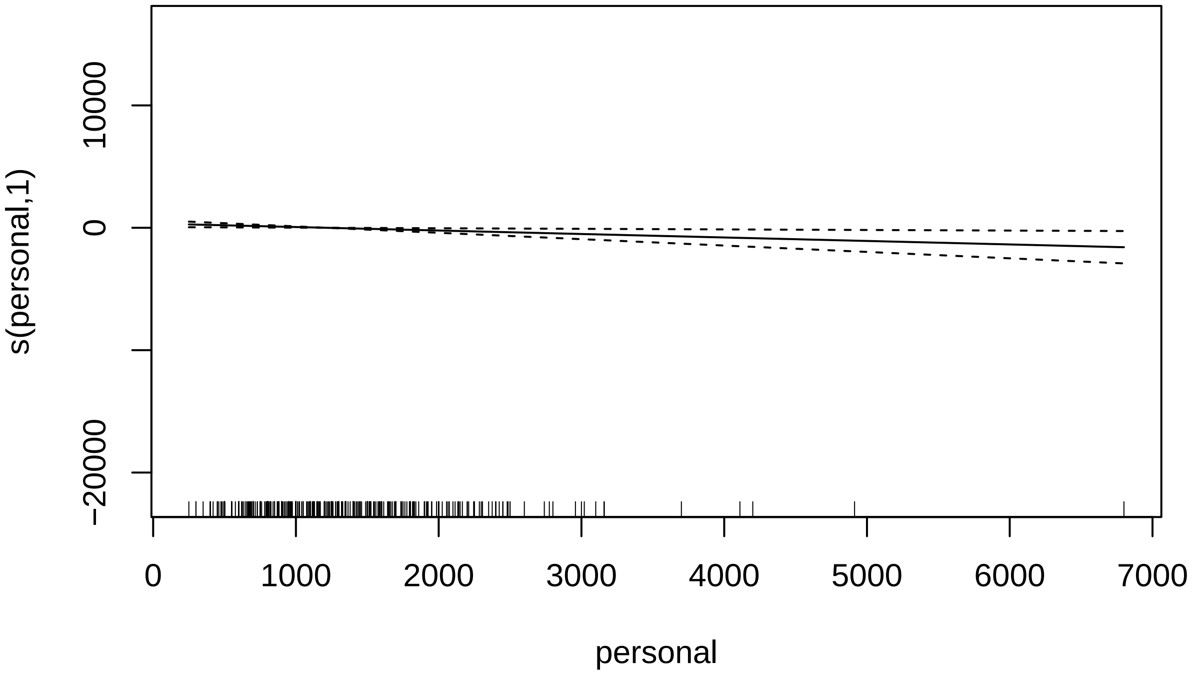

s(ph_d,6.32)
10000
0
−20000
20
40
60
80
100
ph_d

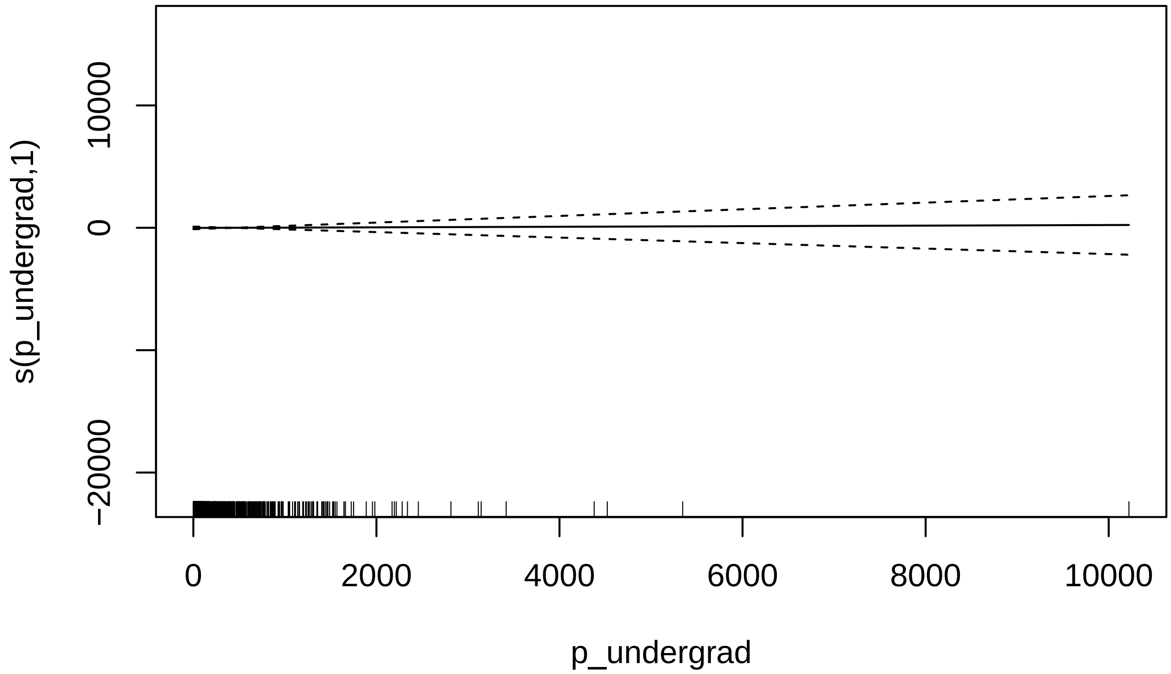

s(grad_rate,4.27)
10000
0
−20000
20
40
60
80
100
120
grad_rate

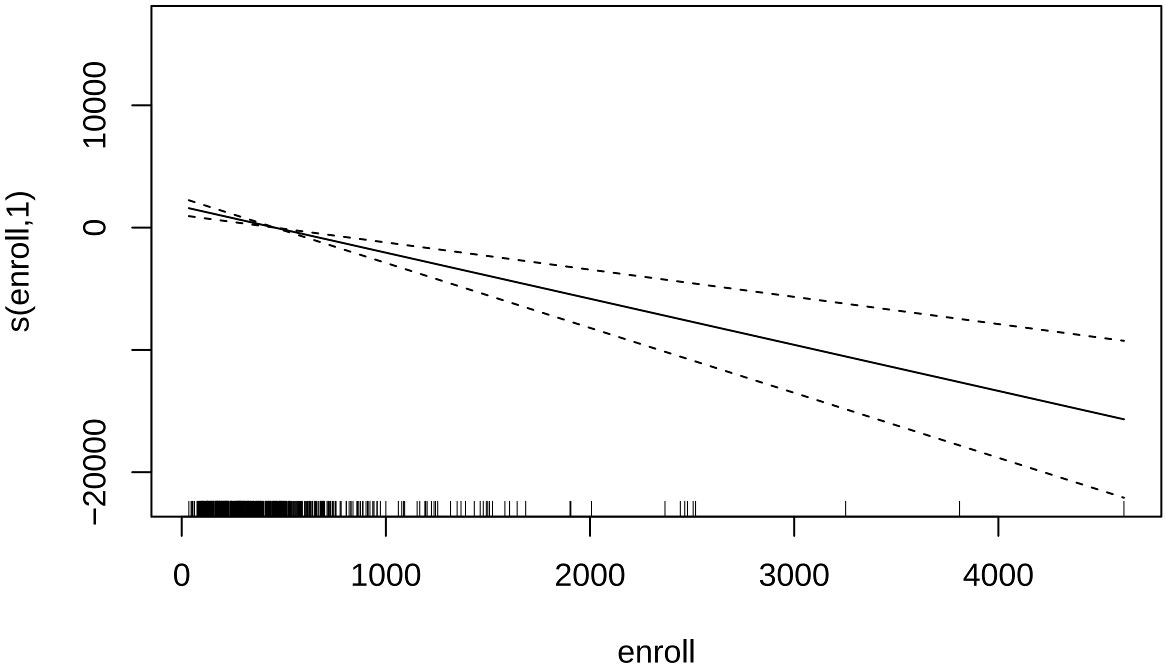

10000
0
−20000
s(books,2.35)
500
1000
1500
2000
books

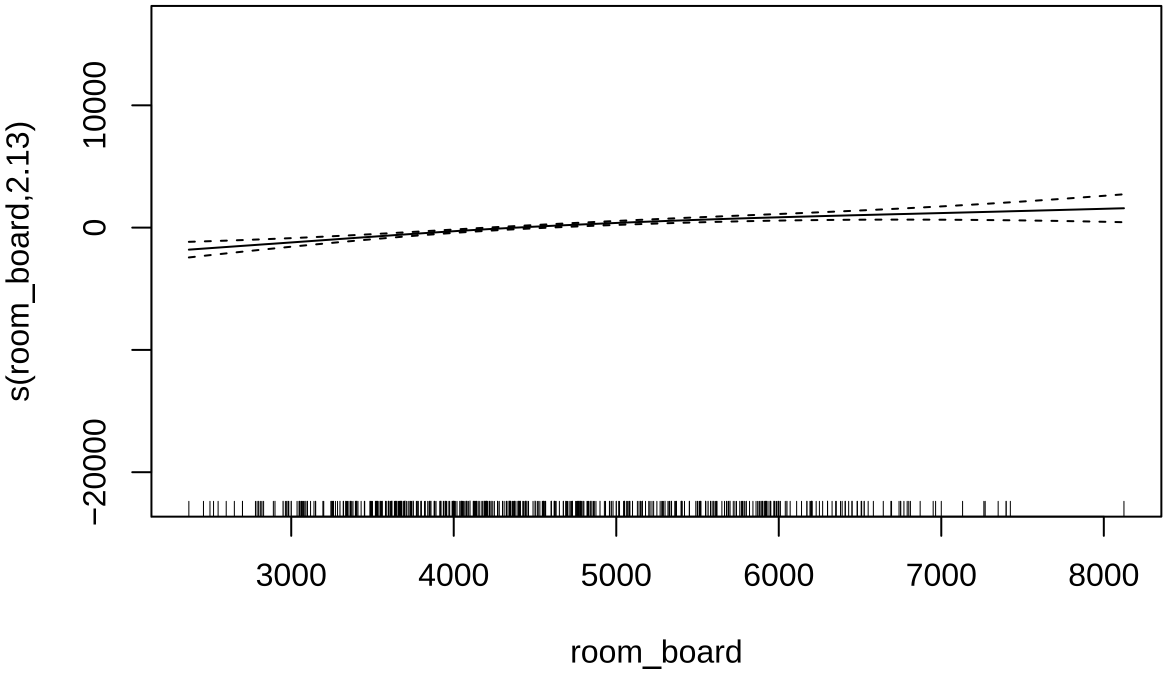

s(top25perc,1)
10000
0
−20000
20
40
60
80
100
top25perc

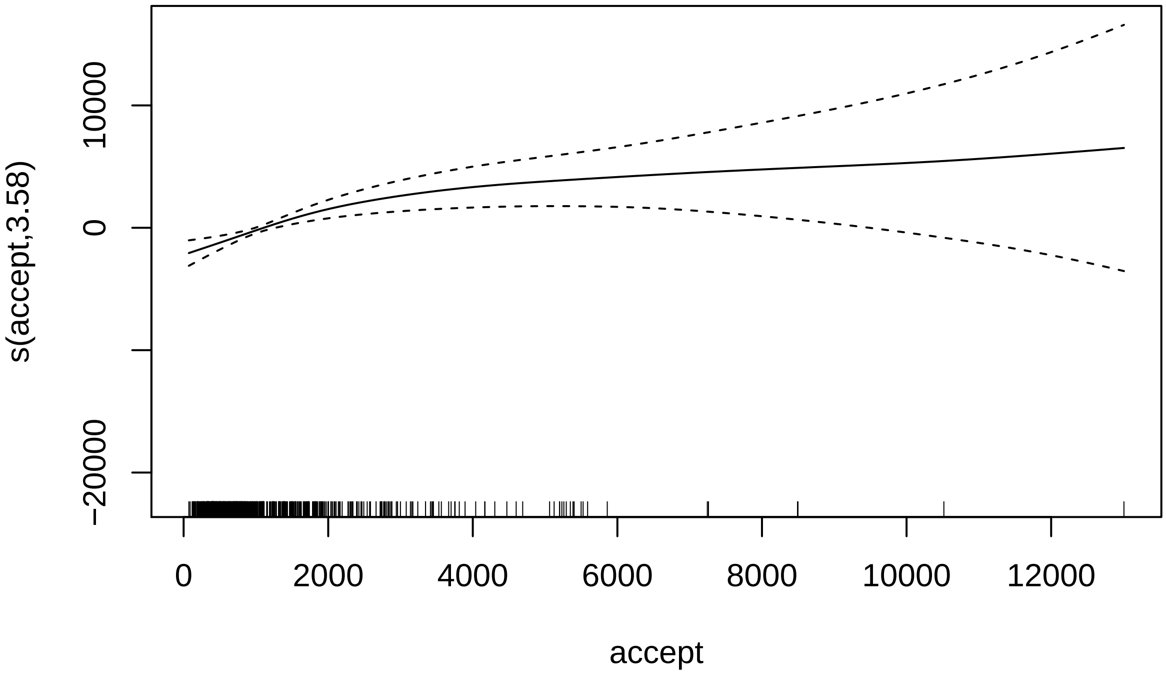

s(s_f_ratio,4.33)
10000
0
−20000
10
20
30
40
s_f_ratio

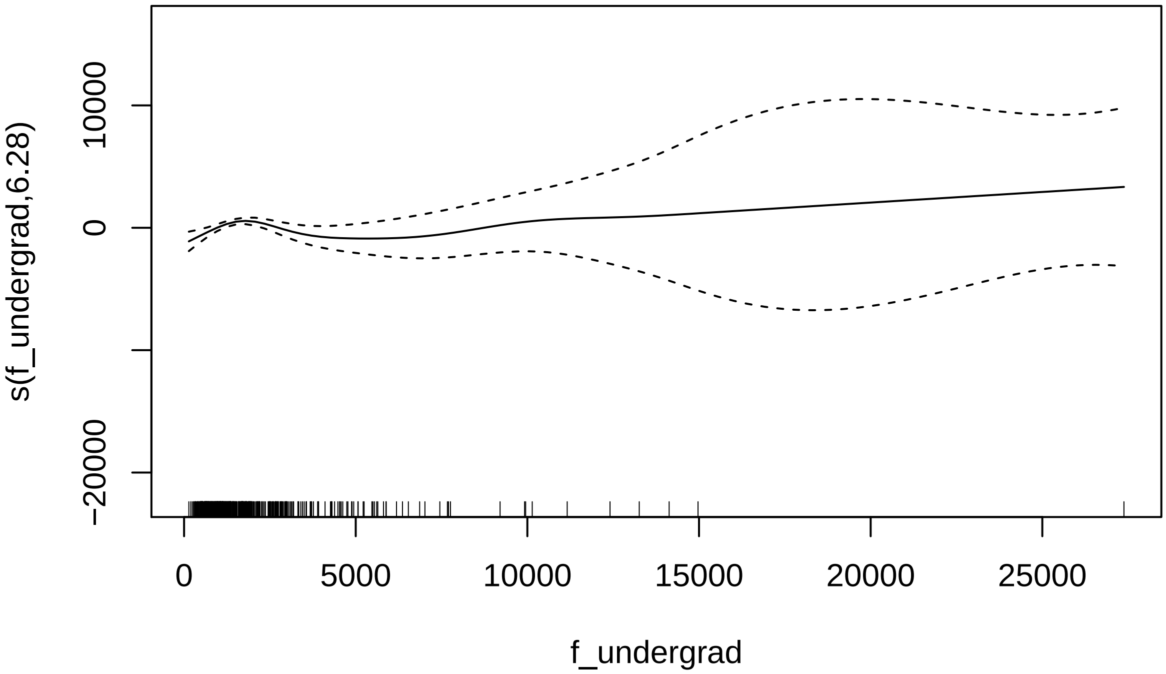

s(personal,1)
10000
0
−20000
0
1000
2000
3000
4000
5000
6000
7000
personal

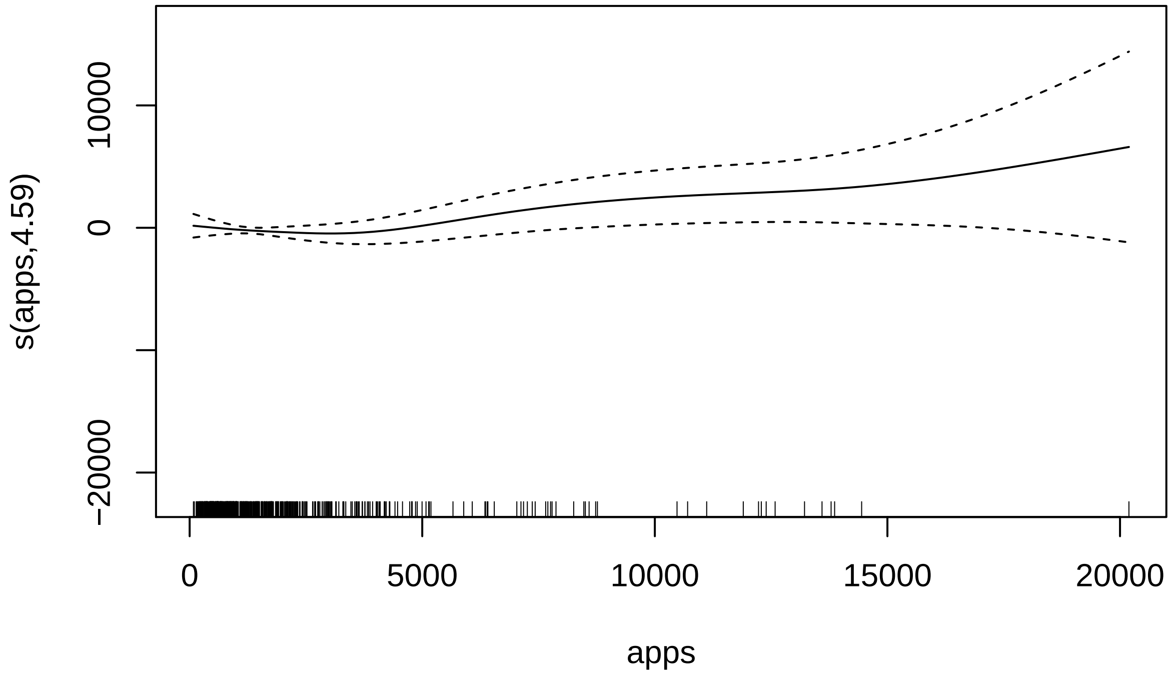

s(p_undergrad,1)
10000
0
−20000
0
2000
4000
6000
8000
10000
p_undergrad

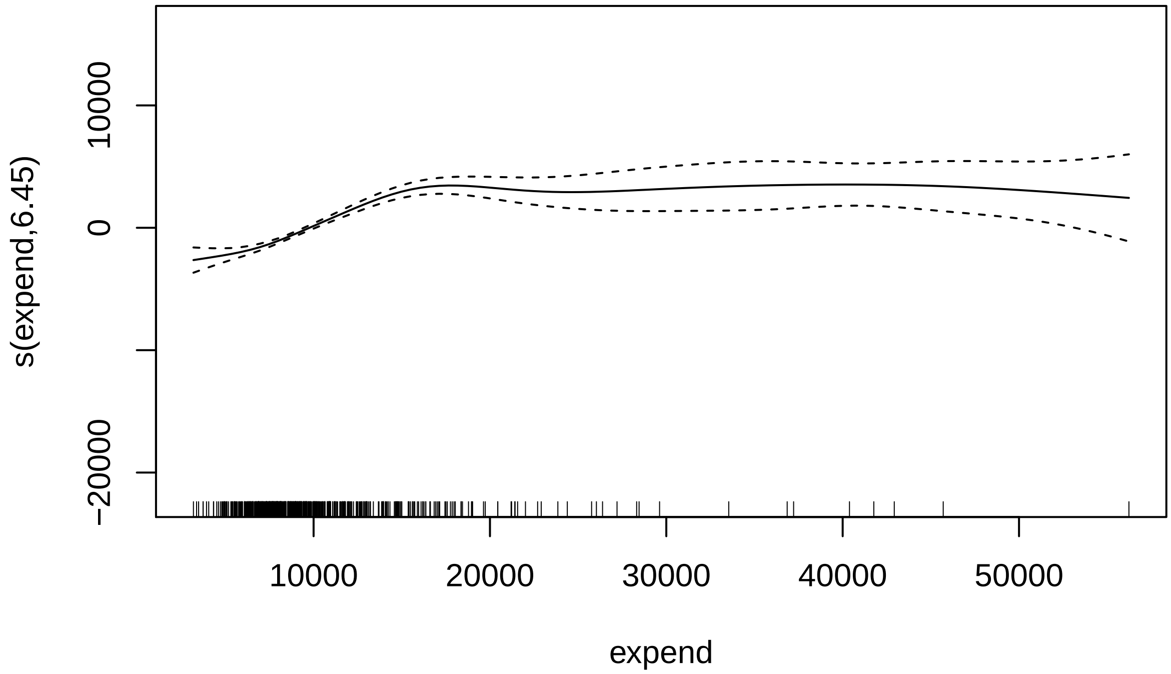

10000
0
−20000
s(enroll,1)
0
1000
2000
3000
4000
enroll


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


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

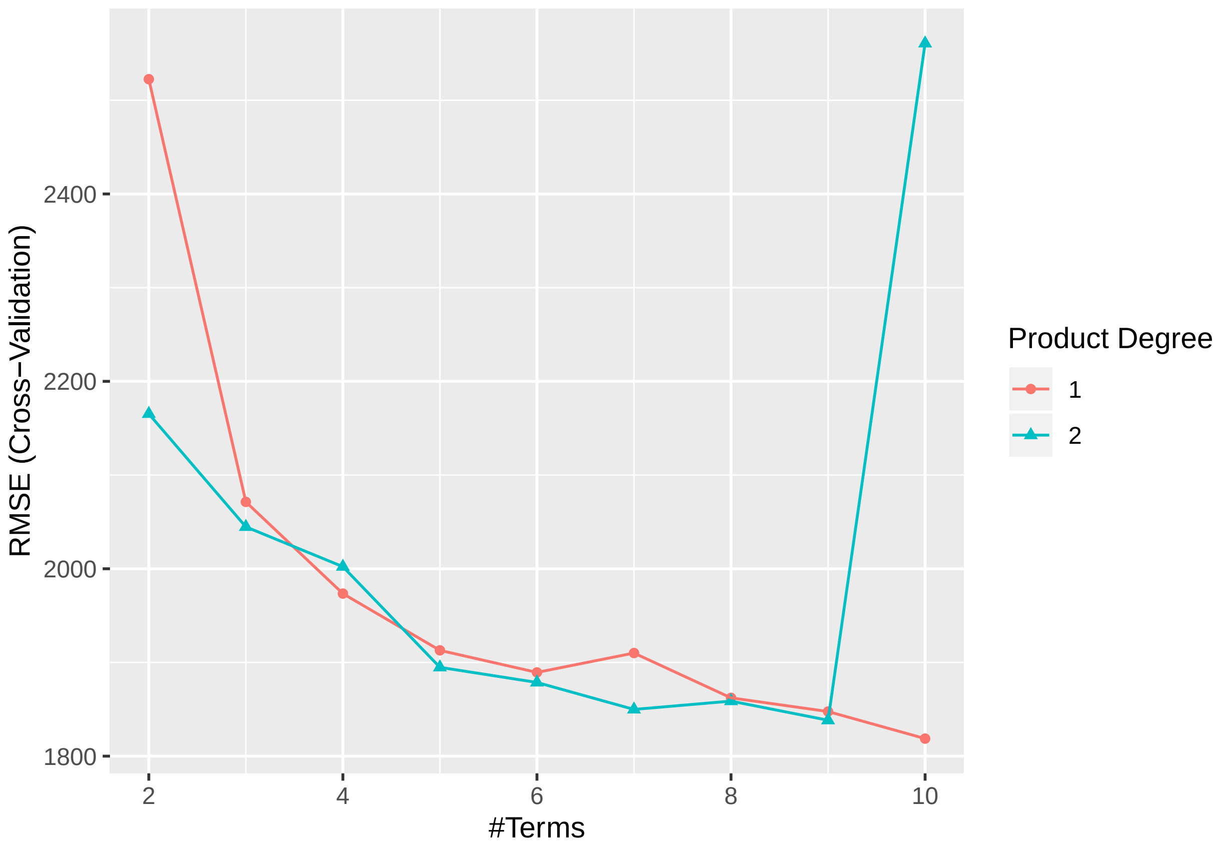

s(apps,4.59)
10000
0
−20000
0
5000
10000
15000
20000
apps



The sum of the degree of freedom is 55.98 (both parametric and non-parametric). The minimized GCV score is 2761951. GAM modeling smoothed out all the 16 variables using splines.

Part D. MARS

```
#Fitting Multivariate adaptive regression splines (MARS)

#Grid searches to figure out tuning parameters.
mars_grid = expand.grid(degree = 1:2,
                        nprune = 2:10)

set.seed(123)
mars_fit = train(x, y,
                 method = "earth",
                 tuneGrid = mars_grid,
                 trControl = control_1)

ggplot(mars_fit)
```



```
mars_fit$finalModel
```

```
## Selected 10 of 29 terms, and 7 of 16 predictors
## Termination condition: Reached nk 33
## Importance: expend, room_board, perc_alumni, accept, f_undergrad, apps, ...
## Number of terms at each degree of interaction: 1 9 (additive model)
## GCV 2968116    RSS 1563128712    GRSq 0.7835269    RSq 0.7971476
```

```
#Mars best tune and final model.
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
```

```r
#Partial dependence plot(pdp) of arbitary variable: room_board
partial_1 = partial(mars_fit,
                    pred.var = c("room_board"),
                    grid.resolution = 10) %>%
  autoplot()
```

The final MARS model has 7 of the 16 predictors: room_board, expend, f_undergrad, perc_alumni, apps, enroll, accept. THe final model MARS model has only selected 10 of the 29 terms. The arbitary variable: room_board was used to create a partial dependence plot (pdp).

Part E. Predicting Columbia's out of state tuition with GAM and MARS.

```r
gam_cumc = predict(gam_fit,
                   newdata = college_cumc)

mars_cumc = predict(mars_fit,
                    newdata = college_cumc)
```

The predicted out of state tuition for Columbia using the GAM model is 17728.51 dollars. The predicted out of state tuition for Columbia using the MARS model is 17469.90 dollars.