P8106 Assignment 2

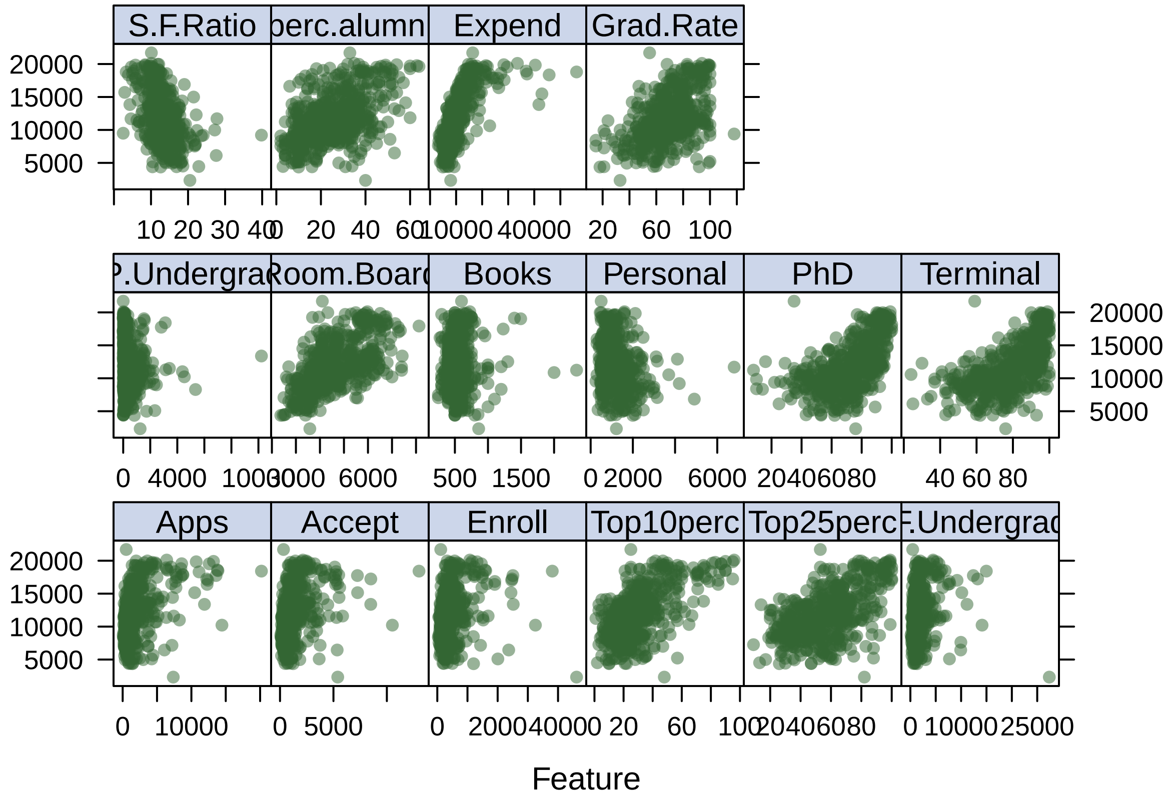
Yihan Feng

2021/2/28

```r
setwd("C:/Users/irene/OneDrive - cumc.columbia.edu/2021 M1 Spring/Data Science 2/HW/hw2")
college.df = read_csv("./College.csv") %>%
  drop_na()
```

(a) Perform exploratory data analysis (e.g., scatter plots of response vs. predictors).

```r
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)
college.df %>%
  dplyr::select(-Outstate, -College) %>%
  featurePlot(., college.df$Outstate, plot = "scatter")
```


S.F.Ratio
perc.alumni
Expend
Grad.Rate
P.Undergrad
Room.Board
Books
Personal
PhD
Terminal
Apps
Accept
Enroll
Top10perc
Top25perc
F.Undergrad
20000
15000
10000
5000
Feature

(b) Fit smoothing spline models using Terminal as the only predictor of Outstate for a range of degrees of freedom, as well as the degree of freedom obtained by generalized cross-validation, and plot the resulting fits. Describe the results obtained.

```
fit.ss = smooth.spline(college.df$Terminal, college.df$Outstate, df = 2)
fit.ss$df
```

1. set degree of freedom as 2

```
## [1] 2.000314
```

```
terminallims = range(college.df$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.2 = ggplot(data = college.df, aes(x = Terminal, y = Outstate)) +
      geom_point(color = "grey") +
      geom_line(aes(x = Terminal, y = pred), data = pred.ss.df,
                color = rgb(.8, .1, .1, 1)) + theme_bw()
```

```
fit.ss = smooth.spline(college.df$Terminal, college.df$Outstate, df = 10)
fit.ss$df
```

2. set degree of freedom as 10

```
## [1] 10.00143
```

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

p.10 = ggplot(data = college.df, aes(x = Terminal, y = Outstate)) +
    geom_point(color = "grey") +
    geom_line(aes(x = Terminal, y = pred), data = pred.ss.df,
              color = rgb(.8, .1, .1, 1)) + theme_bw()
```

```
fit.ss = smooth.spline(college.df$Terminal, college.df$Outstate, df = 20)
fit.ss$df
```

3. set degree of freedom as 20

```
## [1] 20.00251
```

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

p.20 = ggplot(data = college.df, aes(x = Terminal, y = Outstate)) +
    geom_point(color = "grey") +
    geom_line(aes(x = Terminal, y = pred), data = pred.ss.df,
              color = rgb(.8, .1, .1, 1)) + theme_bw()
```

```
fit.ss.cv = smooth.spline(college.df$Terminal, college.df$Outstate, cv = FALSE)
fit.ss.cv$df
```

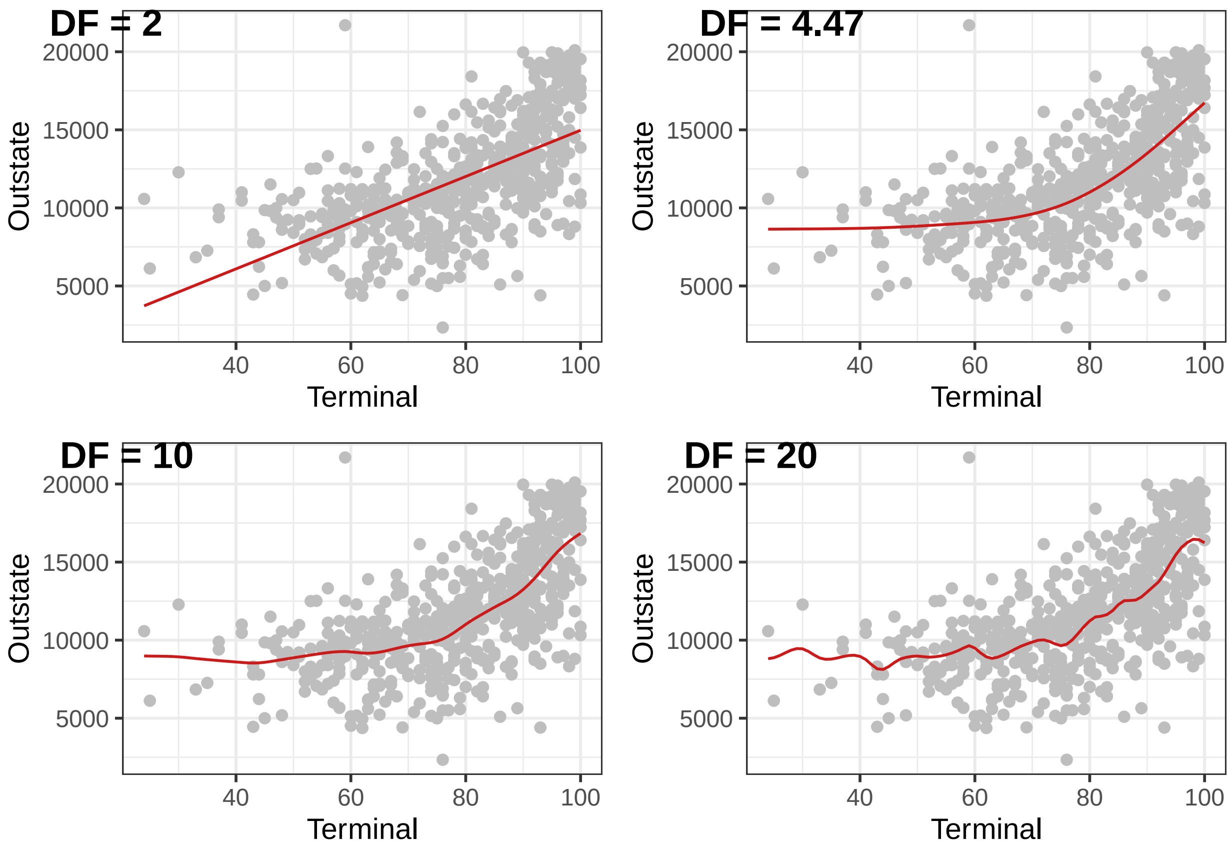
4. degree of freedom obtained by generalized cross-validation.

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

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

p.cv = ggplot(data = college.df, aes(x = Terminal, y = Outstate)) +
      geom_point(color = "grey") +
      geom_line(aes(x = Terminal, y = pred), data = pred.ss.df.cv,
                color = rgb(.8, .1, .1, 1)) + theme_bw()
```

```
ggpubr::ggarrange(p.2, p.cv, p.10, p.20,
                  labels = c("DF = 2", "DF = 4.47", "DF = 10", "DF = 20"),
                  ncol = 2, nrow = 2)
```



According to the three plots, when the degree of freedom is larger, the line is much wiggly; when the degree of freedom is smaller, the line tends to be linear. The degree of freedom obtained from cross-validation (4.4686294), shows a smooth curve.

(c) Fit a generalized additive model (GAM) using all the predictors. Plot the results and explain your findings.

```
x = model.matrix(Outstate ~ .,college.df)[,-1]
y = college.df$Outstate

ctrl1 = trainControl(method = "cv", number = 10)
set.seed(1)
gam.fit = train(x, y,
                method = "gam",
                tuneGrid = data.frame(method = "GCV.Cp",
                                      select = c(TRUE,FALSE)),
                 trControl = ctrl1)
gam.fit$bestTune
```

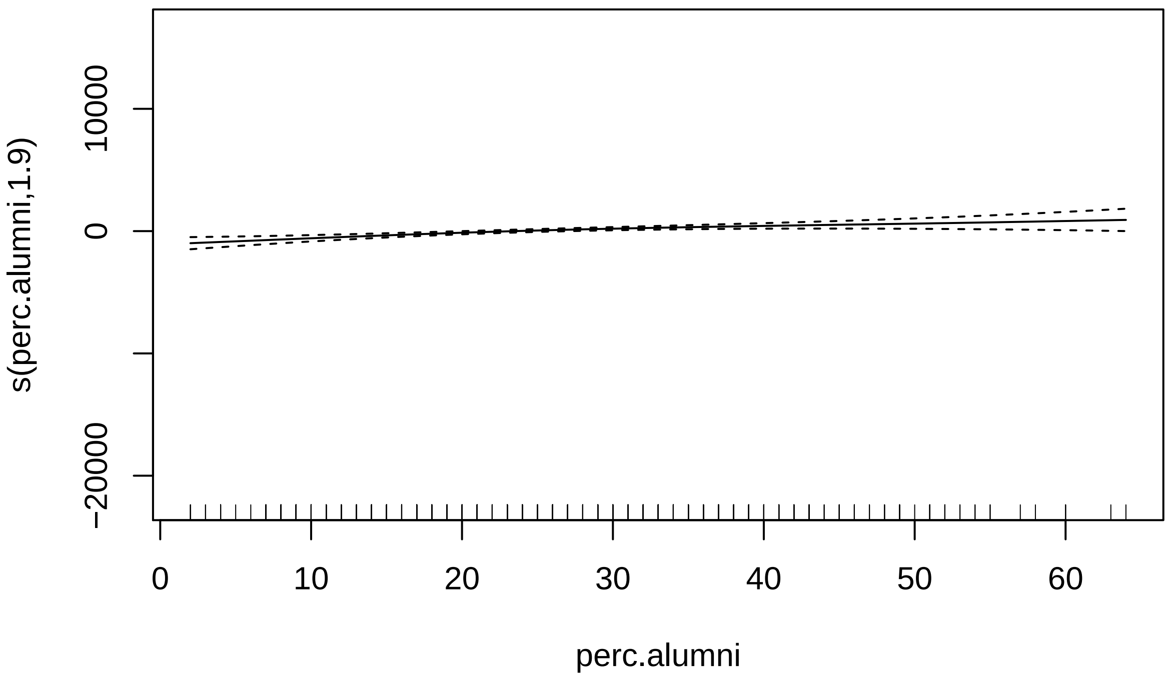
```
##   select method
## 1  FALSE GCV.Cp
```

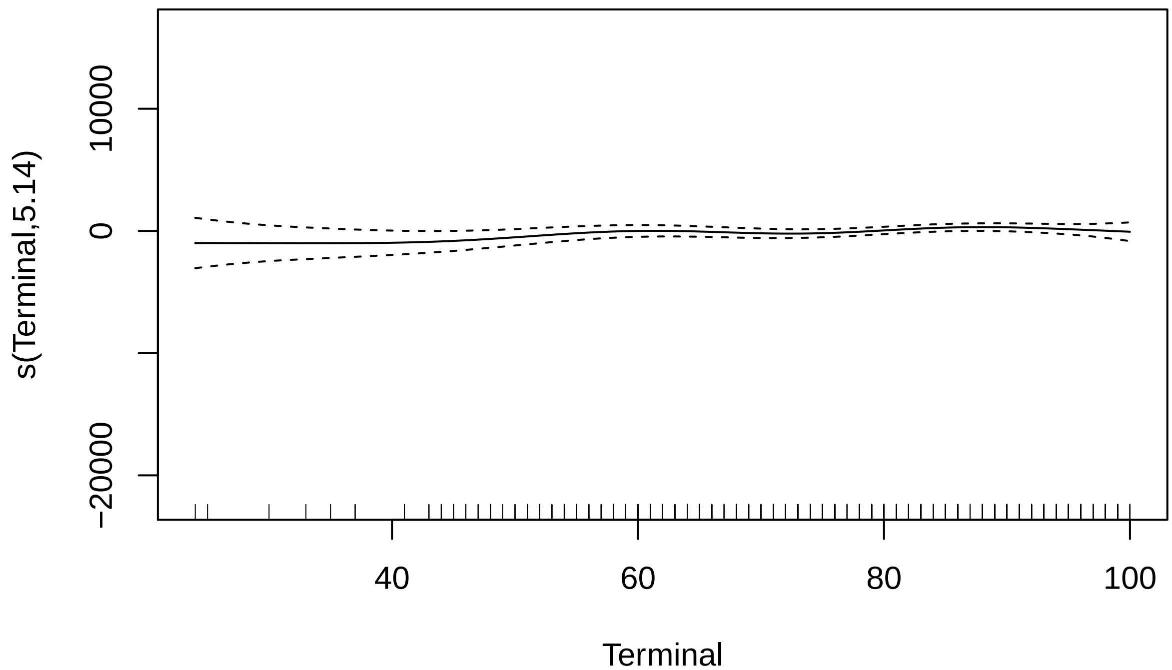
```
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc.alumni) + s(Terminal) + s(Top10perc) + s(PhD) +
##     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
```

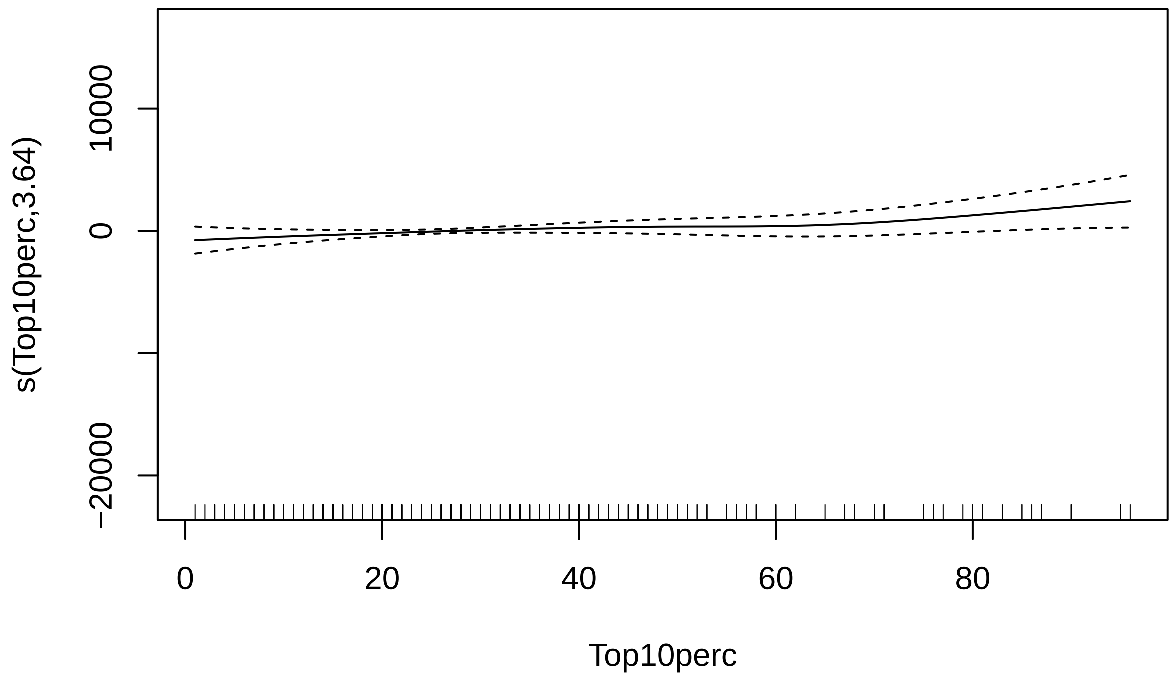
According to the final model:

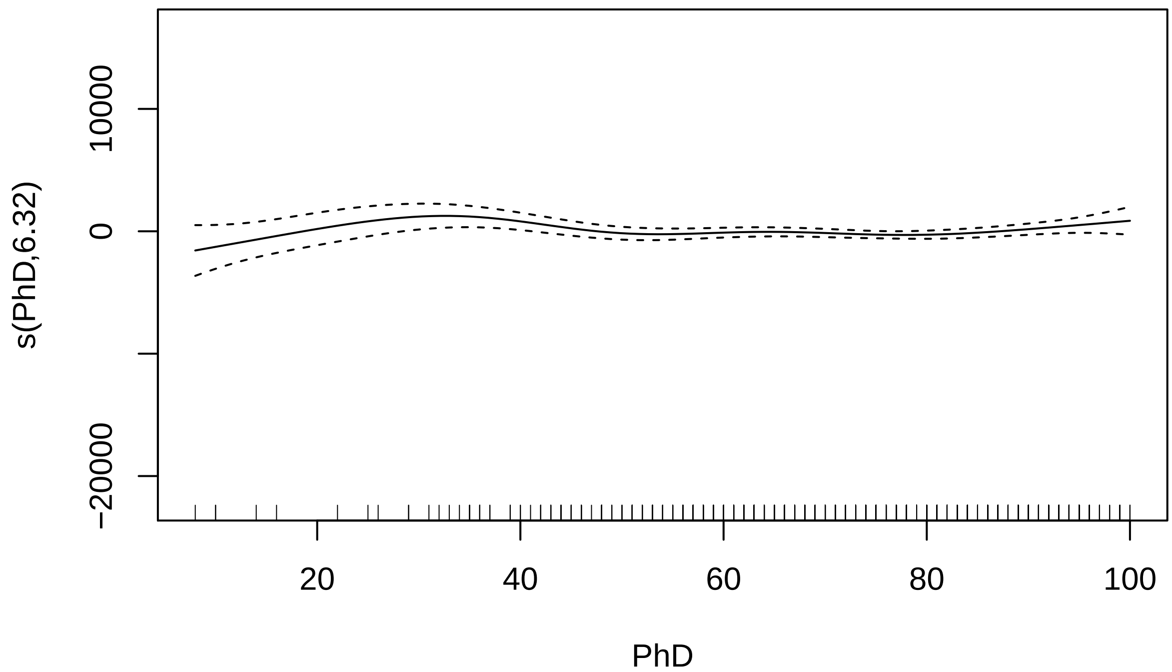
```
plot(gam.fit$finalModel)
```


s(perc.alumni,1.9)
10000
0
−20000
0
10
20
30
40
50
60
perc.alumni

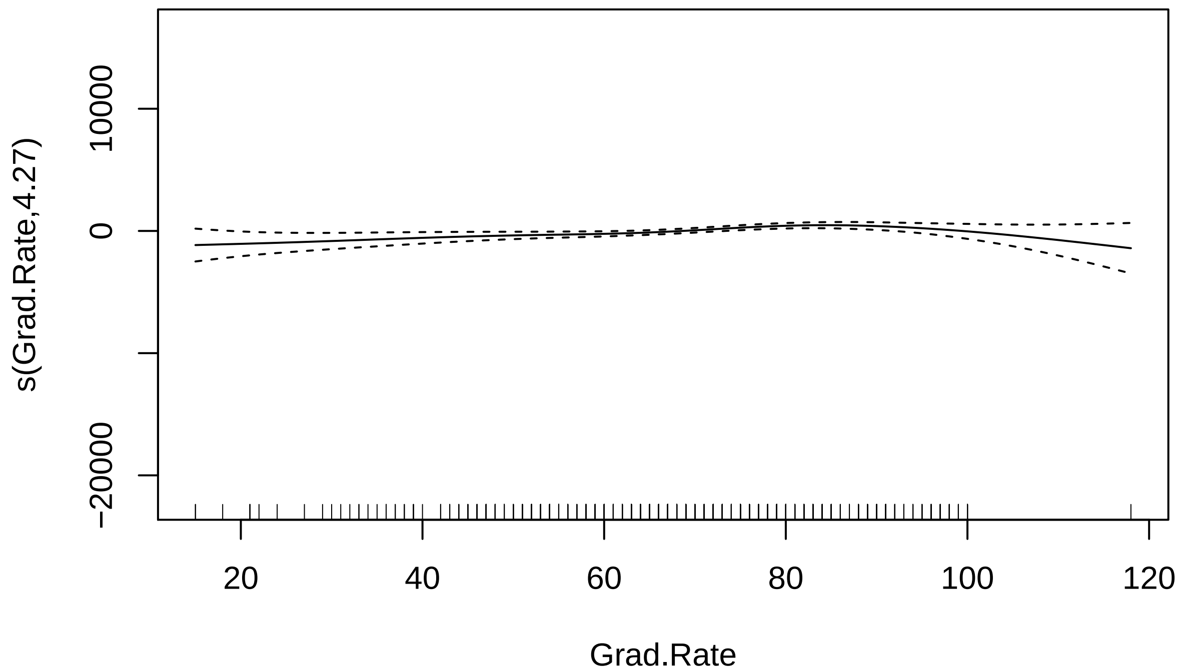


s(Terminal,5.14)
10000
0
−20000
40
60
80
100
Terminal

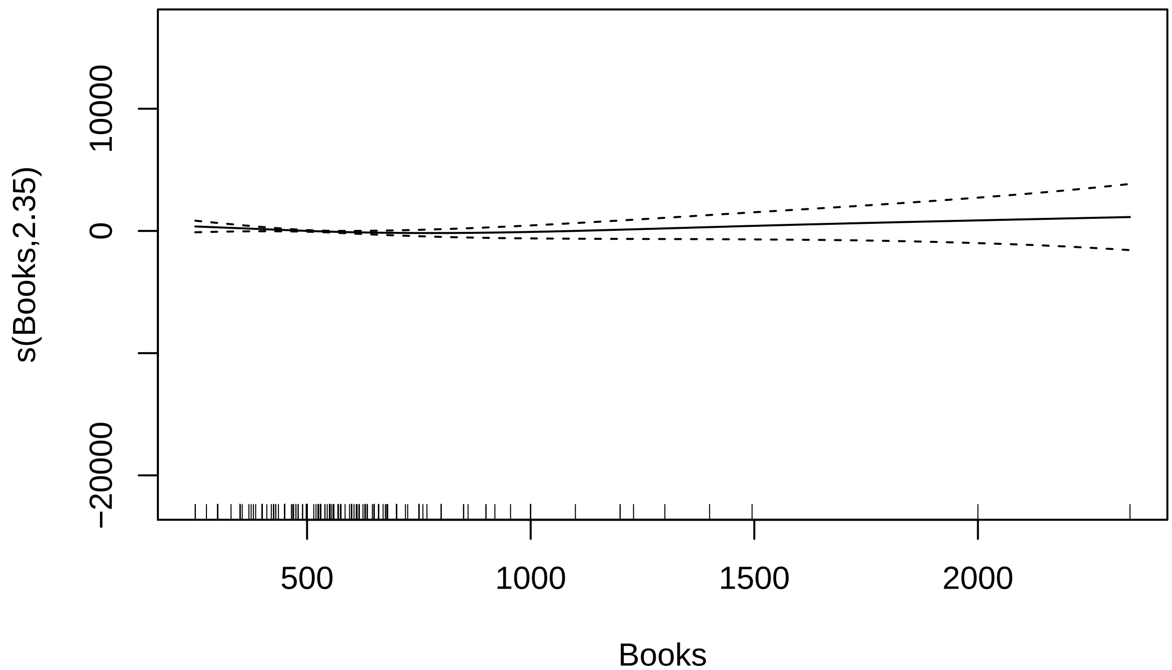

s(Top10perc,3.64)
10000
0
−20000
0
20
40
60
80
Top10perc



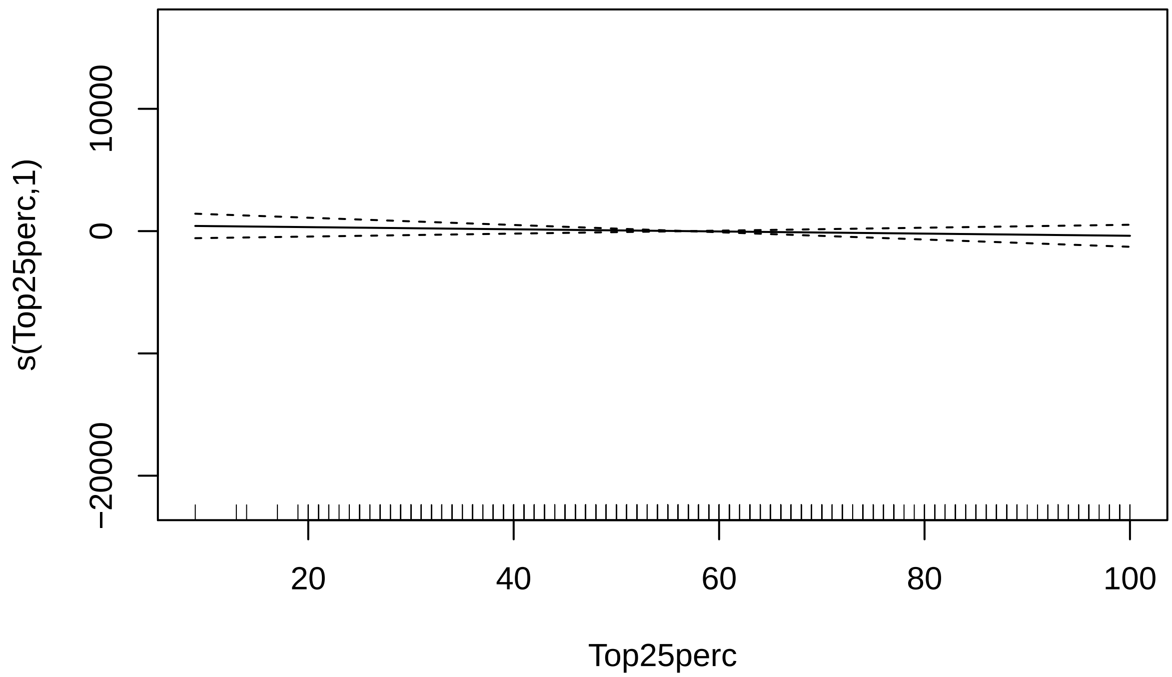
s(PhD,6.32)
10000
0
−20000
20
40
60
80
100
PhD

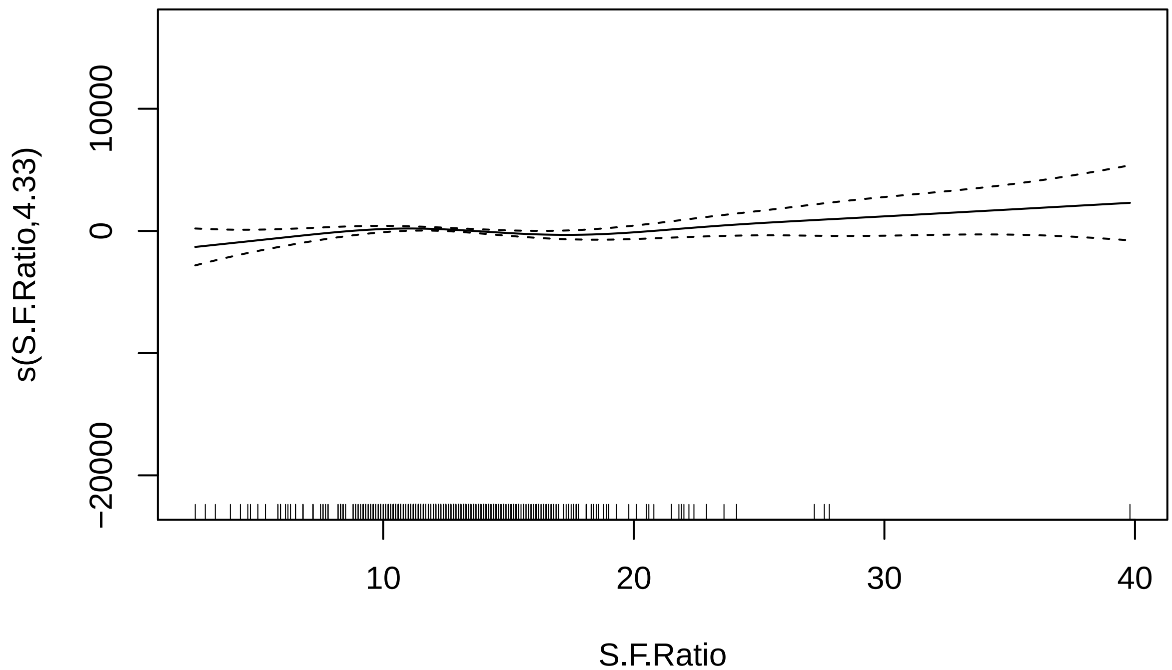


s(Grad.Rate,4.27)
10000
0
−20000
20
40
60
80
100
120
Grad.Rate

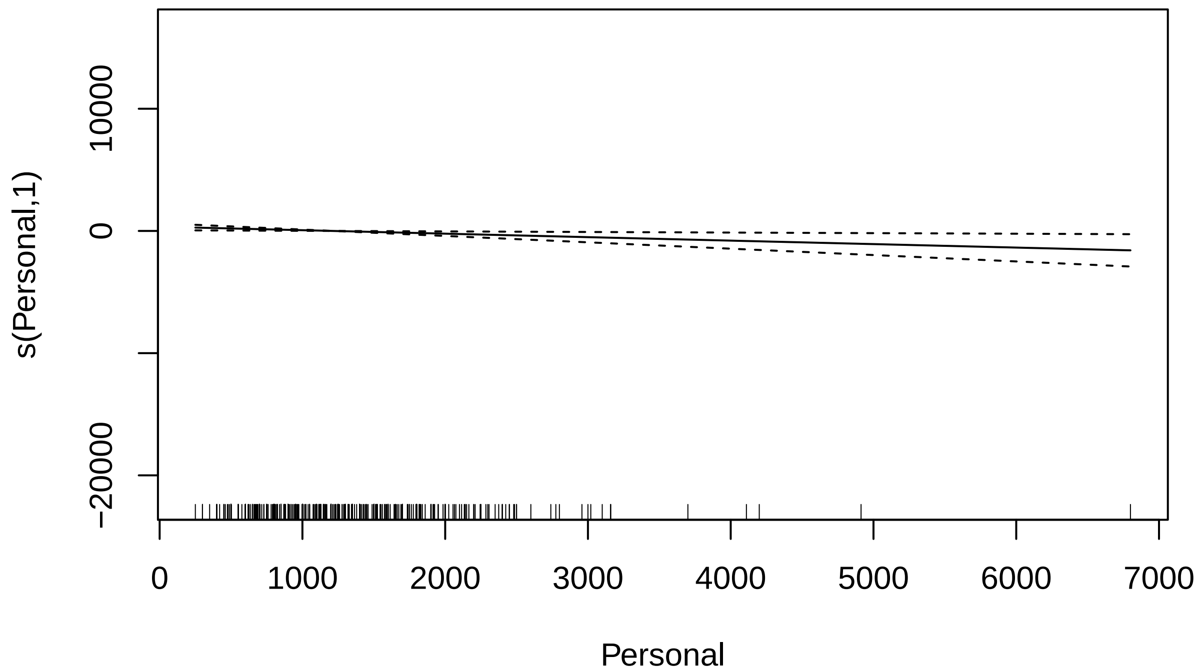


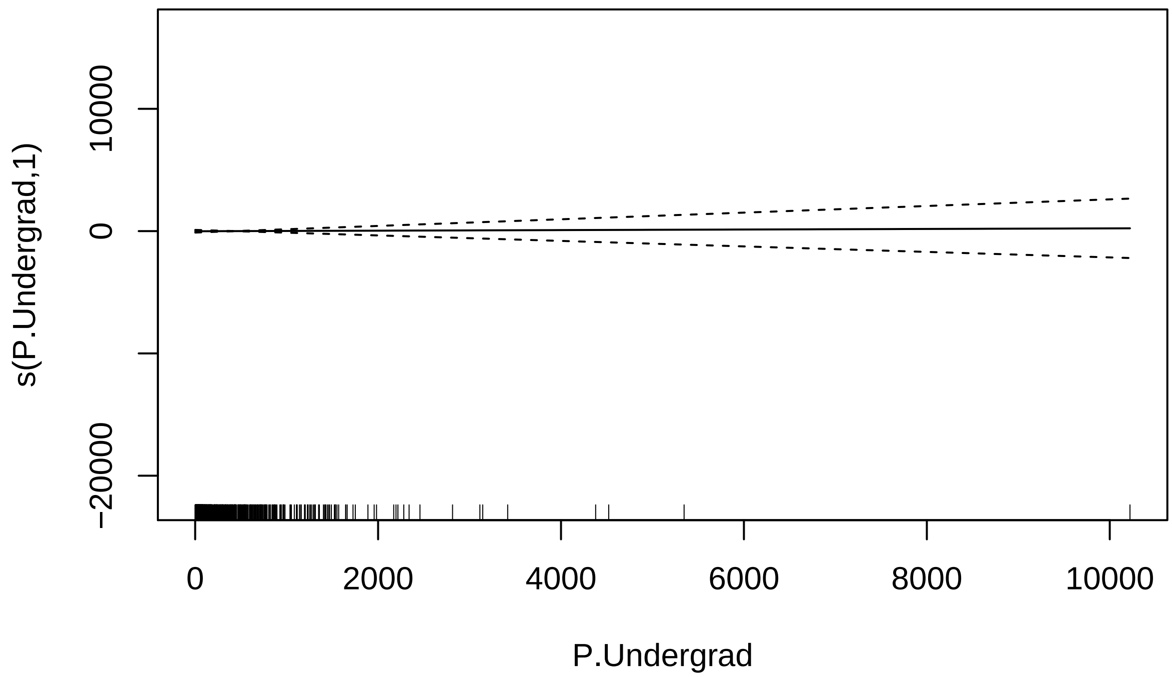
10000
0
−20000
s(Books,2.35)
500
1000
1500
2000
Books

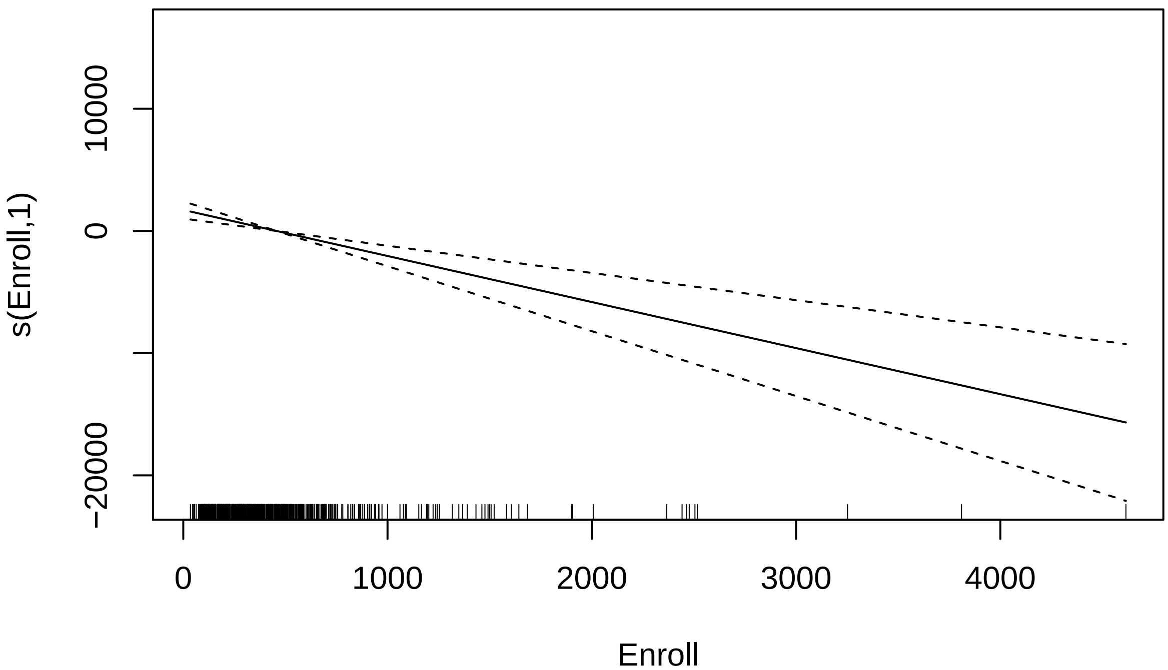

s(Top25perc,1)
10000
0
−20000
20
40
60
80
100
Top25perc



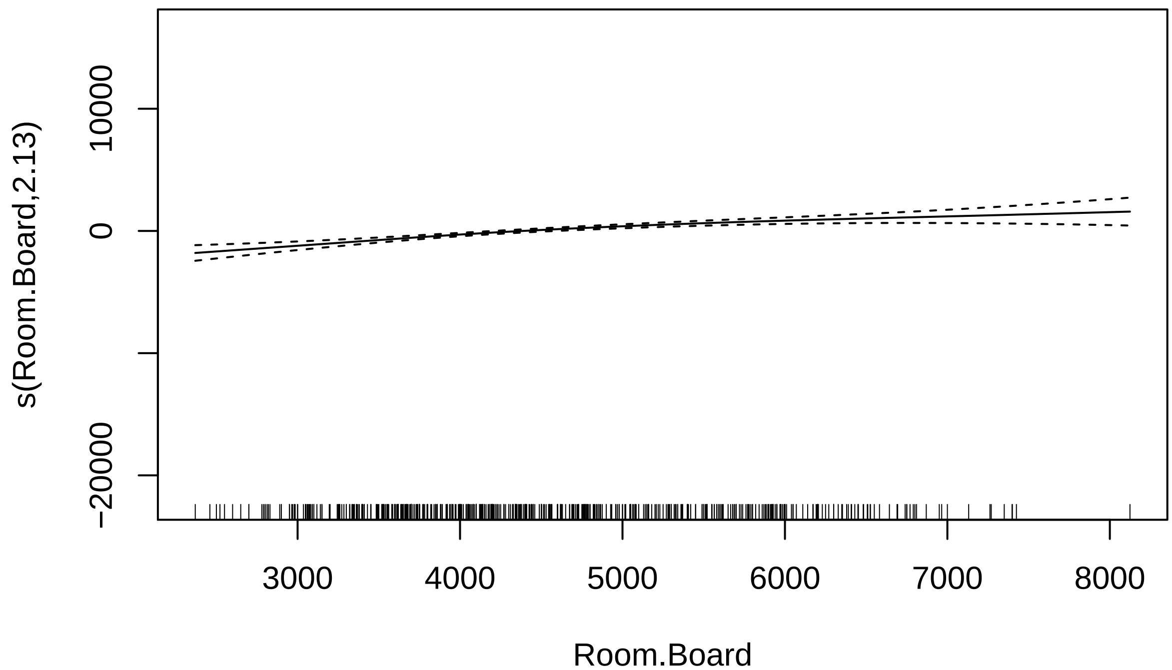
s(S.F.Ratio,4.33)
10000
0
−20000
10
20
30
40
S.F.Ratio


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

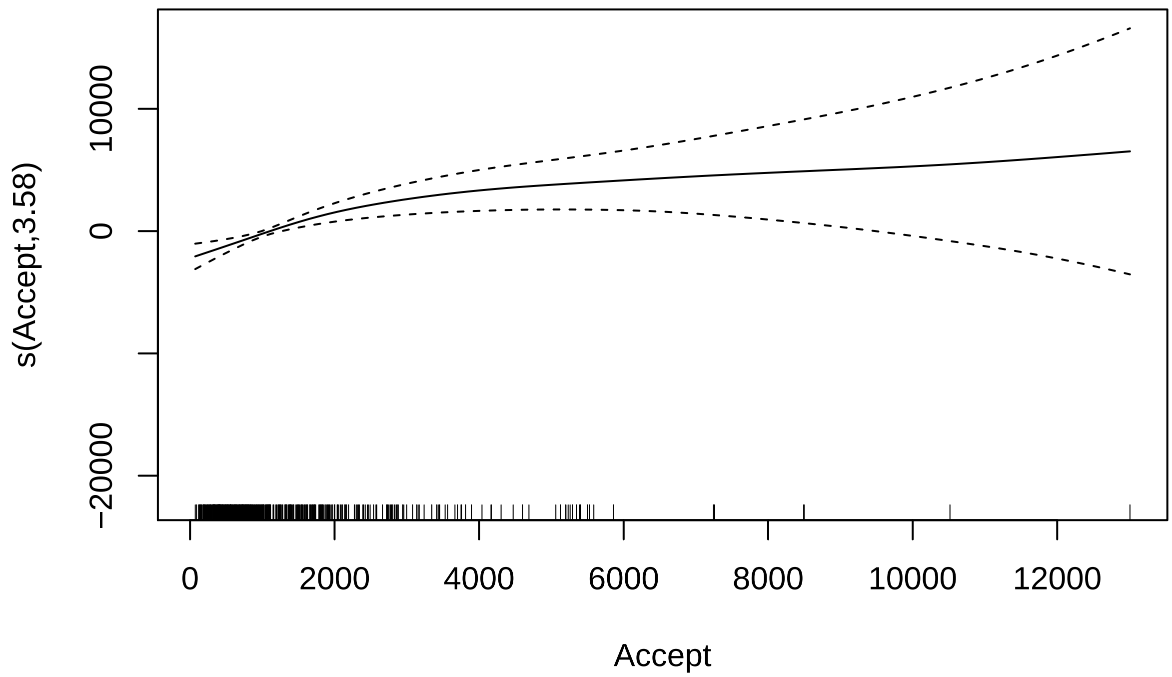

10000
0
−20000
s(P.Undergrad,1)
0
2000
4000
6000
8000
10000
P.Undergrad



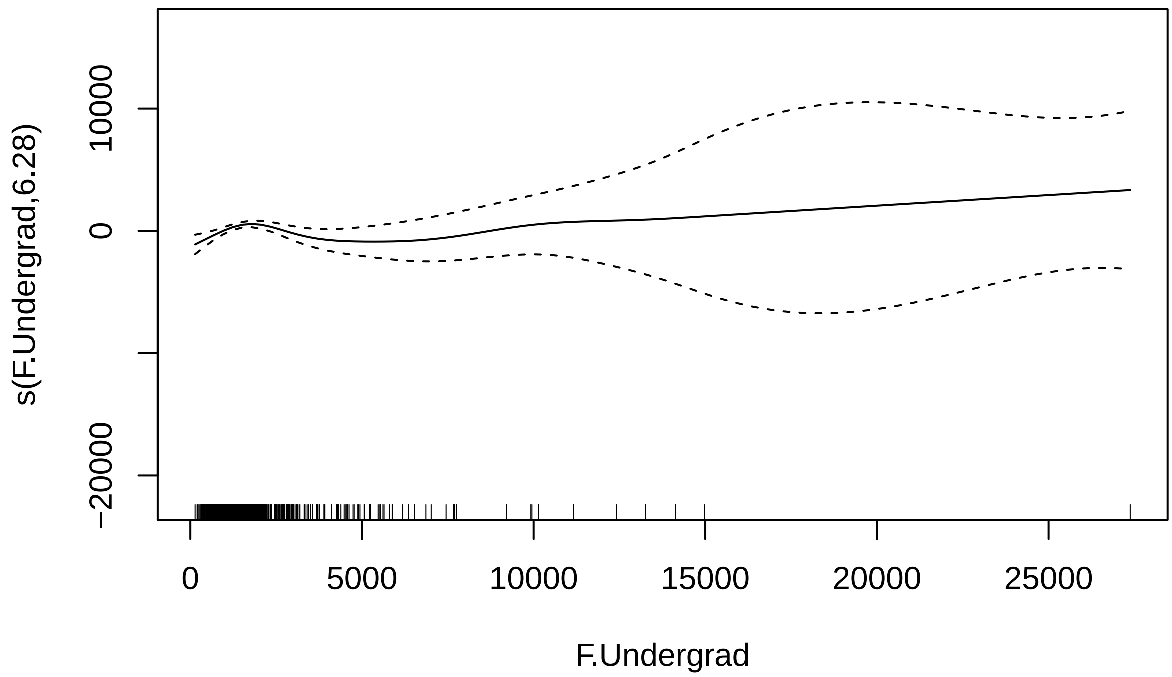
s(Enroll,1)
10000
0
−20000
0
1000
2000
3000
4000
Enroll

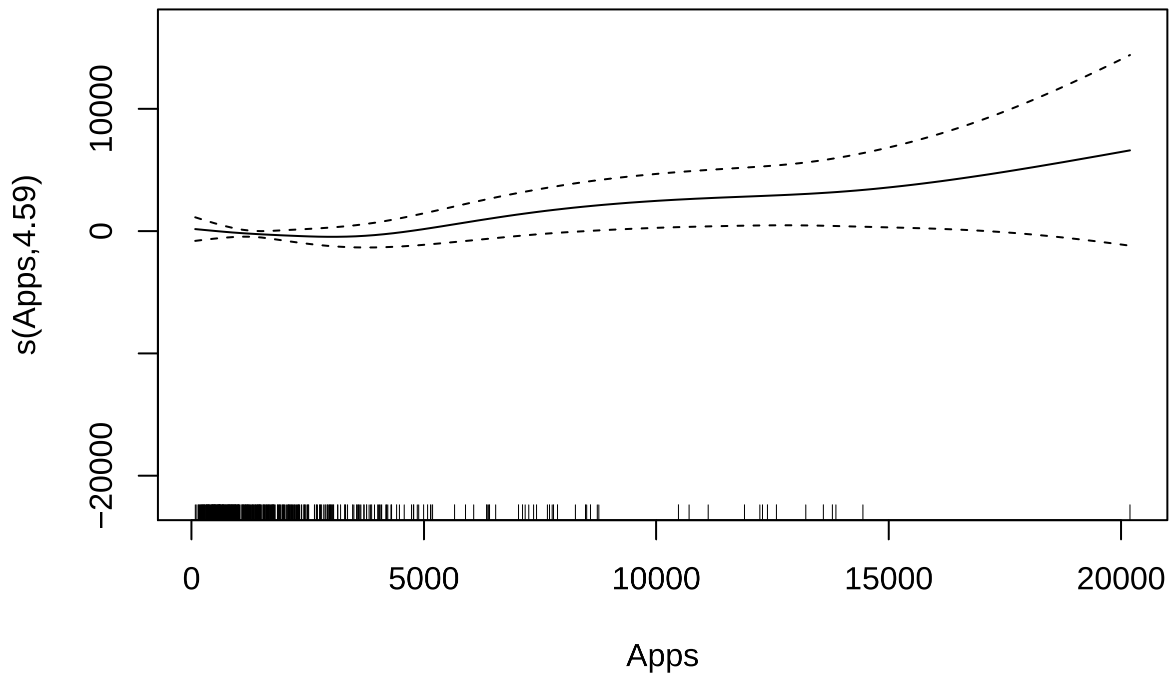


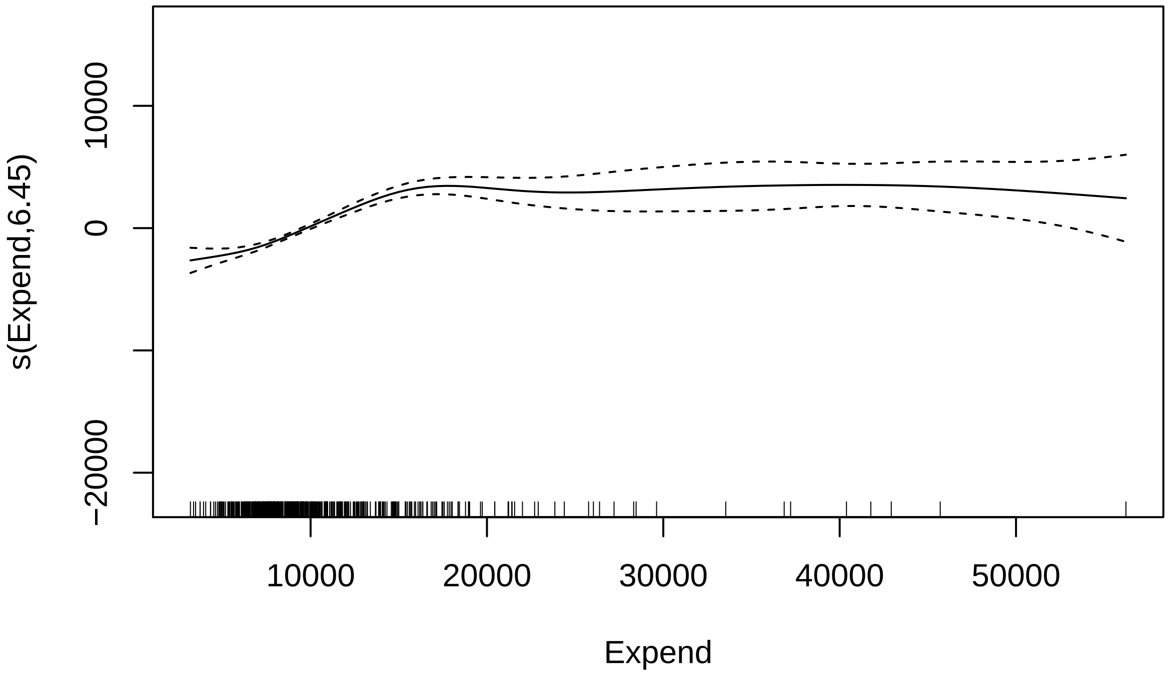
s(Room.Board,2.13)
10000
0
−20000
3000
4000
5000
6000
7000
8000
Room.Board



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



According to the final model, I found that the best model has "select = FALSE", and "method = GCV.Cp". And all predictors has the spline function. However, using the caret method, we may lose a significant amount of flexibility in `mgcv`, such as interactions.

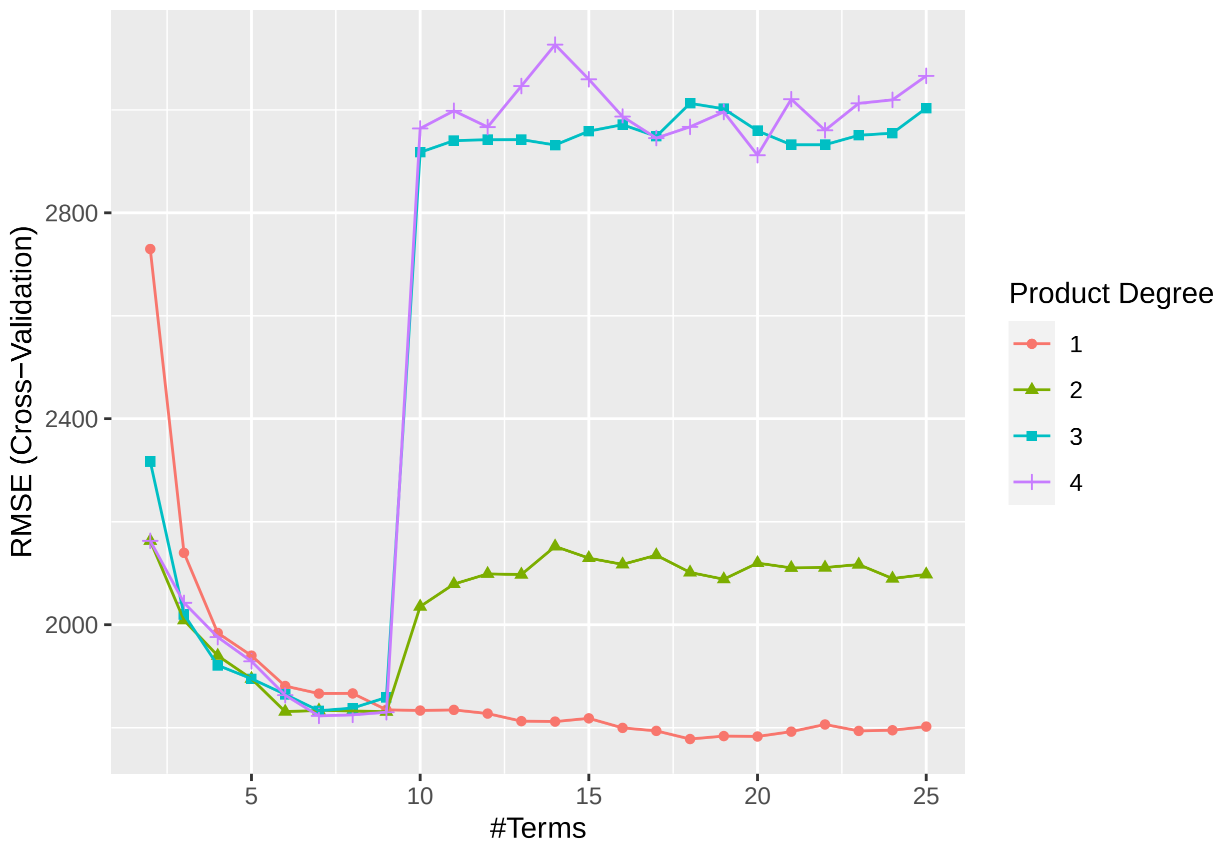
(d) Train a multivariate adaptive regression spline (MARS) model using all the predictors. Report the final model. Present the partial dependence plot of an arbitrary predictor in your final model.

```
mars_grid <- expand.grid(degree = 1:4,
                        nprune = 2:25)

set.seed(1)

mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)
```




```
mars.fit$bestTune
```

```
##    nprune degree
## 17     18      1
```

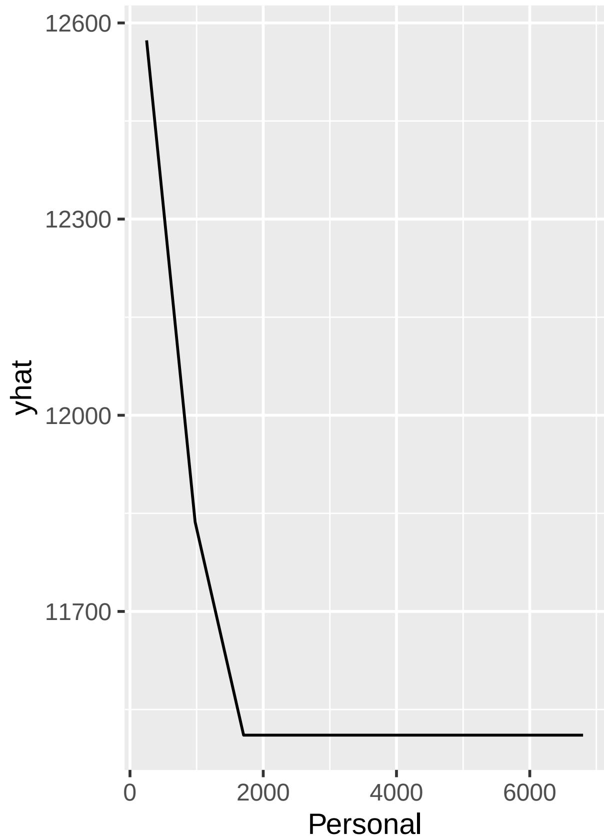
```
coef(mars.fit$finalModel)
```

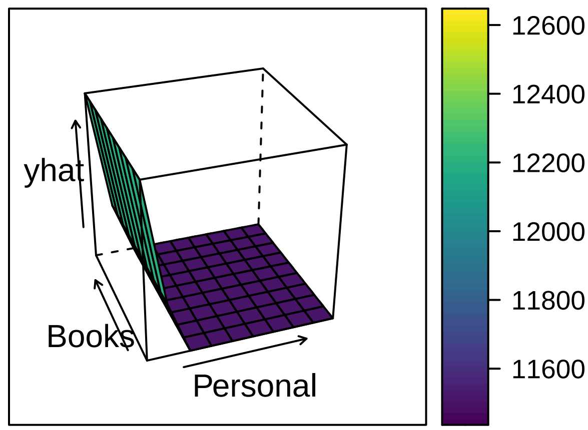
```
##                                 (Intercept)
```

```
##                              10416.4815354
##                            h(Expend-15622)
##                                 -0.7303018
##                         h(4440-Room.Board)
##                                 -1.1712495
##                            h(95-Grad.Rate)
##                                -25.2562986
##                        h(F.Undergrad-1350)
##                                 -0.3228898
##                        h(1350-F.Undergrad)
##                                 -1.4271250
##                          h(21-perc.alumni)
##                                -68.6772542
##                               h(Apps-3767)
##                                  0.3729090
##                           h(1300-Personal)
##                                  1.0120010
##                              h(903-Enroll)
##                                  4.3464161
##                             h(2165-Accept)
##                                 -1.8782536
##                 CollegeBennington College
##                               6101.1592594
## CollegeWentworth Institute of Technology
##                              -6092.7298267
##                CollegeLivingstone College
##                              -5965.2462884
##                             h(Expend-5970)
##                                  0.7331585
##               CollegeCreighton University
##                              -5992.8123315
##                 CollegeTrinity University
##                              -5695.1532326
##   CollegeArkansas College (Lyon College)
##                              -5542.0120027
```

```
p1 = pdp::partial(mars.fit, pred.var = c("Personal"),
                  grid.resolution = 10) %>%
  autoplot()
p2 <- pdp::partial(mars.fit,
                   pred.var = c("Personal", "Books"),
                    grid.resolution = 10) %>%
  pdp::plotPartial(levelplot = FALSE,
                   zlab = "yhat",
                   drape = TRUE,
                   screen = list(z = 20, x = -60))

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






```r
resamp = resamples(list(mars = mars.fit,
                        gam = gam.fit))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: mars, gam
## Number of resamples: 10
##
## MAE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## mars 1206.940 1311.868 1359.676 1369.621 1414.591 1529.858    0
## gam  1110.792 1273.871 1362.840 1355.597 1405.497 1597.107    0
##
## RMSE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## mars 1614.034 1672.231 1759.103 1778.018 1896.806 1979.644    0
## gam  1495.091 1656.657 1704.097 1772.331 1913.426 2143.493    0
##
## Rsquared
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## mars 0.7099505 0.7554377 0.7751064 0.7734302 0.7996594 0.8157176    0
## gam  0.6578161 0.7671896 0.7910314 0.7757679 0.8143682 0.8236840    0
```