P8106_HW2_yh3554_w/Correction

Yi Huang

2023-03-12

Contents

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

Data Science II Homework 2

In this exercise, we build nonlinear models using the "College" data. The dataset contains statistics for 565 US Colleges from a previous issue of US News and World Report. The response variable is the out-of-state tuition (Outstate). The predictors are

- Apps: Number of applications received
- Accept: Number of applications accepted
- Enroll: Number of new students enrolled
- Top10perc: Pct. new students from top 10% of H.S. class
- Top25perc: Pct. new students from top 25% of H.S. class
- F.Undergrad: Number of fulltime undergraduates
- P.Undergrad: Number of parttime undergraduates
- Room.Board: Room and board costs
- Books: Estimated book costs
- Personal: Estimated personal spending
- PhD: Pct. of faculty with Ph.D.'s
- Terminal: Pct. of faculty with terminal degree
- S.F.Ratio: Student/faculty ratio
- perc.alumni: Pct. alumni who donate
- Expend: Instructional expenditure per student
- Grad.Rate: Graduation rate

Partition the dataset into two parts: training data (80%) and test data (20%).

(a)

Fit smoothing spline models using perc.alumni 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 smoothing spline models with different df

```
# set seed for reproducibility
set.seed(123)

# load data
dat <- read.csv("data/College.csv")
dat <- na.omit(dat)
head(dat)
```

```
##                         College Apps Accept Enroll Top10perc Top25perc
## 1 Abilene Christian University 1660   1232    721        23        52
```

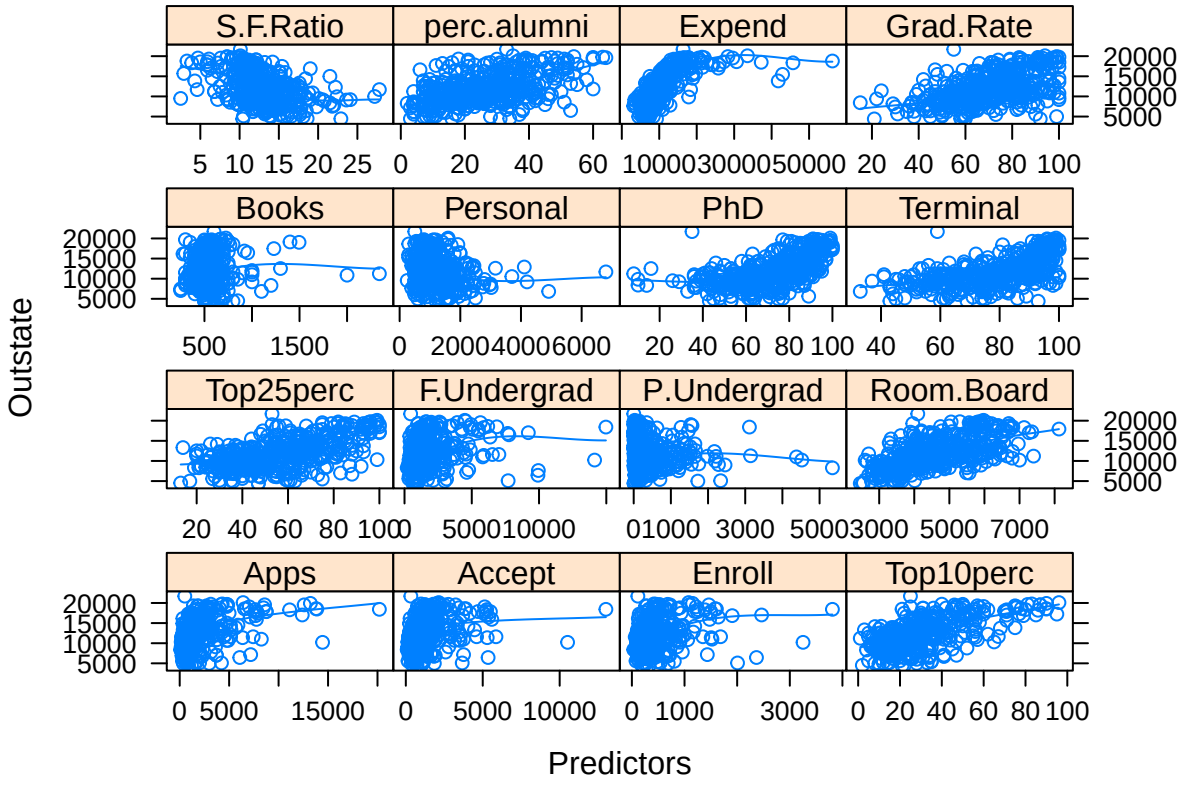
```
## 2           Adelphi University 2186   1924    512        16        29
## 3              Adrian College 1428   1097    336        22        50
## 4          Agnes Scott College  417    349    137        60        89
## 5    Alaska Pacific University  193    146     55        16        44
## 6            Albertson College  587    479    158        38        62
##   F.Undergrad P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1        2885         537     7440       3300   450     2200  70       78
## 2        2683        1227    12280       6450   750     1500  29       30
## 3        1036          99    11250       3750   400     1165  53       66
## 4         510          63    12960       5450   450      875  92       97
## 5         249         869     7560       4120   800     1500  76       72
## 6         678          41    13500       3335   500      675  67       73
##   S.F.Ratio perc.alumni Expend Grad.Rate
## 1      18.1          12   7041        60
## 2      12.2          16  10527        56
## 3      12.9          30   8735        54
## 4       7.7          37  19016        59
## 5      11.9           2  10922        15
## 6       9.4          11   9727        55
```

```
summary(dat)
```

```
##    College               Apps           Accept          Enroll
##  Length:565         Min.   :   81   Min.   :   72   Min.   :  35.0
##  Class :character   1st Qu.:  619   1st Qu.:  501   1st Qu.: 206.0
##  Mode  :character   Median : 1133   Median :  859   Median : 328.0
##                     Mean   : 1978   Mean   : 1306   Mean   : 456.9
##                     3rd Qu.: 2186   3rd Qu.: 1580   3rd Qu.: 520.0
##                     Max.   :20192   Max.   :13007   Max.   :4615.0
##    Top10perc       Top25perc       F.Undergrad     P.Undergrad
##  Min.   : 1.00   Min.   :  9.00   Min.   :  139   Min.   :    1
##  1st Qu.:17.00   1st Qu.: 42.00   1st Qu.:  840   1st Qu.:   63
##  Median :25.00   Median : 55.00   Median : 1274   Median :  207
##  Mean   :29.33   Mean   : 56.96   Mean   : 1872   Mean   :  434
##  3rd Qu.:36.00   3rd Qu.: 70.00   3rd Qu.: 2018   3rd Qu.:  541
##  Max.   :96.00   Max.   :100.00   Max.   :27378   Max.   :10221
##     Outstate       Room.Board       Books           Personal
##  Min.   : 2340   Min.   :2370   Min.   : 250.0   Min.   : 250
##  1st Qu.: 9100   1st Qu.:3736   1st Qu.: 450.0   1st Qu.: 800
##  Median :11200   Median :4400   Median : 500.0   Median :1100
##  Mean   :11802   Mean   :4586   Mean   : 547.5   Mean   :1214
##  3rd Qu.:13970   3rd Qu.:5400   3rd Qu.: 600.0   3rd Qu.:1500
##  Max.   :21700   Max.   :8124   Max.   :2340.0   Max.   :6800
##       PhD            Terminal        S.F.Ratio      perc.alumni
##  Min.   :  8.00   Min.   : 24.00   Min.   : 2.50   Min.   : 2.00
##  1st Qu.: 60.00   1st Qu.: 68.00   1st Qu.:11.10   1st Qu.:16.00
##  Median : 73.00   Median : 81.00   Median :12.70   Median :25.00
##  Mean   : 71.09   Mean   : 78.53   Mean   :12.95   Mean   :25.89
##  3rd Qu.: 85.00   3rd Qu.: 92.00   3rd Qu.:14.50   3rd Qu.:34.00
##  Max.   :100.00   Max.   :100.00   Max.   :39.80   Max.   :64.00
##      Expend        Grad.Rate
##  Min.   : 3186   Min.   : 15
##  1st Qu.: 7477   1st Qu.: 58
##  Median : 8954   Median : 69
##  Mean   :10486   Mean   : 69
```

```
##  3rd Qu.:11625   3rd Qu.: 81
##  Max.   :56233   Max.   :118
```

```
# specify rows of training data (80% of the dataset)
train_rows <- createDataPartition(dat$Outstate,
                                  p = 0.8,
                                  list = F)

# training data
dat_train <- dat[train_rows, ]
x <- dat_train %>% select(-College, -Outstate)
y <- dat_train$Outstate

# test data
dat_test <- dat[-train_rows, ]
x2 <- dat_test %>% select(-College, -Outstate)
y2 <- dat_test$Outstate

# resampling method 10-fold cross-validation
ctrl1 <- trainControl(method = "cv", number = 10)

# scatter plot
featurePlot(x,y,
            plot = "scatter",
            span = 0.5,
            labels = c("Predictors", "Outstate"),
            type = c("p", "smooth"),
            layout = c(4,4))
```



```
# fit smoothing spline model
fit.ss_df3 <- smooth.spline(dat_train$perc.alumni, dat_train$Outstate, df = 3)
fit.ss_df5 <- smooth.spline(dat_train$perc.alumni, dat_train$Outstate, df = 5)
fit.ss_df8 <- smooth.spline(dat_train$perc.alumni, dat_train$Outstate, df = 8)
```

Fit smoothing spline models with df obtained by generalized cross-validation

```
set.seed(123)

# fit smoothing spline model with df obtained by generalized cross-validation
fit.ss_cv <- smooth.spline(dat_train$perc.alumni, dat_train$Outstate)

# retrieve df obtained by generalized cross-validation
fit.ss_cv$df
```

```
## [1] 2.00025
```

```
fit.ss_cv$lambda
```

```
## [1] 2477.678
```

The degree of freedom obtained by generalized cross-validation is 2.0002.

Plot the result fittings

```
range(dat$perc.alumni)
```

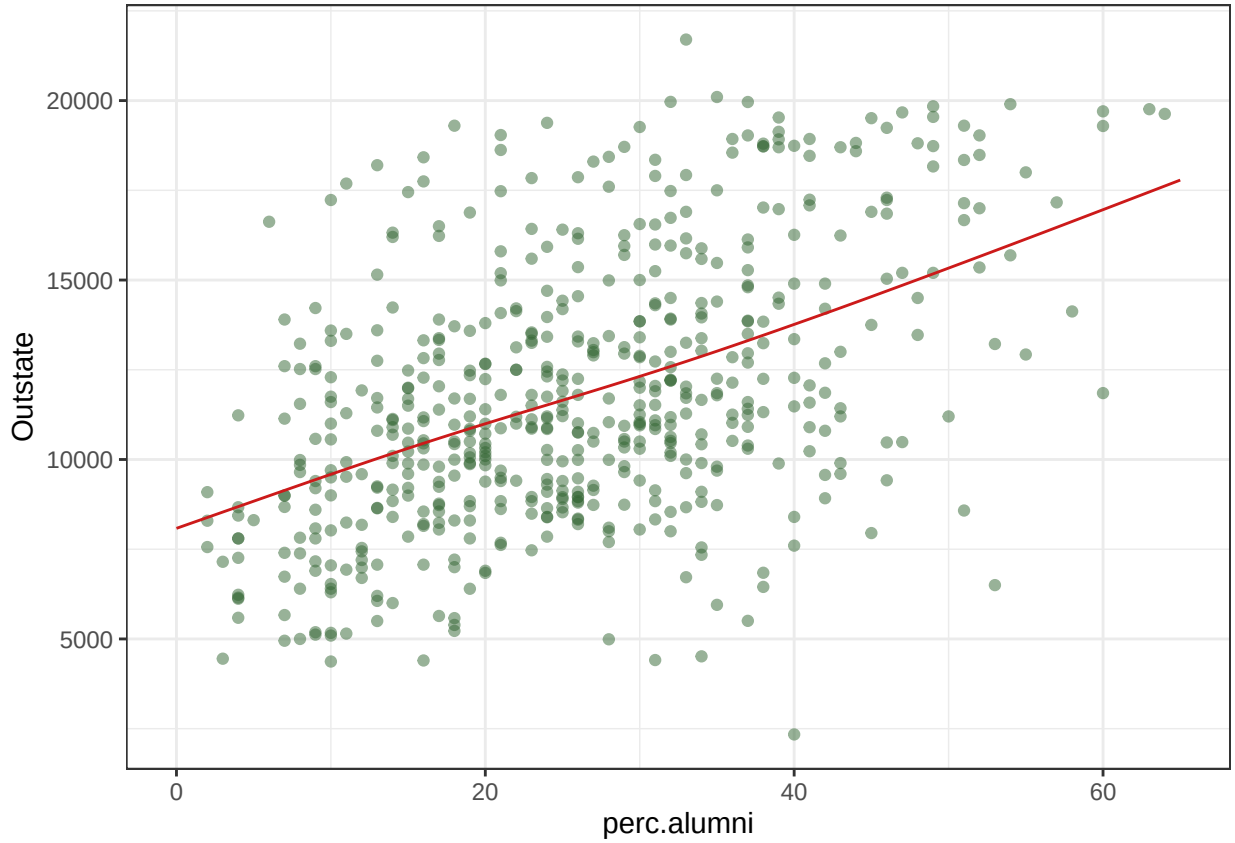
```
## [1]  2 64
```

```
# Note that the range of pgg45 is [2,64], and this is only for
# illustrating fitted curve beyond the boundary knots
perc.alumni.grid <- seq(from = 0, to = 65, by = 1)

# df = 3
pred.ss_df3 <- predict(fit.ss_df3,
                       x = perc.alumni.grid)

pred.ss.df_3 <- data.frame(pred = pred.ss_df3$y,
                           perc.alumni = perc.alumni.grid)
# df = 5
pred.ss_df5 <- predict(fit.ss_df5,
                       x = perc.alumni.grid)

pred.ss.df_5 <- data.frame(pred = pred.ss_df5$y,
                           perc.alumni = perc.alumni.grid)

# df = 8
pred.ss_df8 <- predict(fit.ss_df8,
                       x = perc.alumni.grid)

pred.ss.df_8 <- data.frame(pred = pred.ss_df8$y,
                           perc.alumni = perc.alumni.grid)

# df obtained by generalized cross-validation
pred.ss.cv <- predict(fit.ss_cv,
```

```
                    x = perc.alumni.grid)

pred.ss.df_cv <- data.frame(pred = pred.ss.cv$y,
                            perc.alumni = perc.alumni.grid)
# create scatter plot object 'p' of the data points
# perc.alumni on the x-axis and Outstate on the y-axis
p <- ggplot(data = dat, aes(x = perc.alumni, y = Outstate)) +
  geom_point(color = rgb(.2, .4, .2, .5))
```

Plot for df = 3

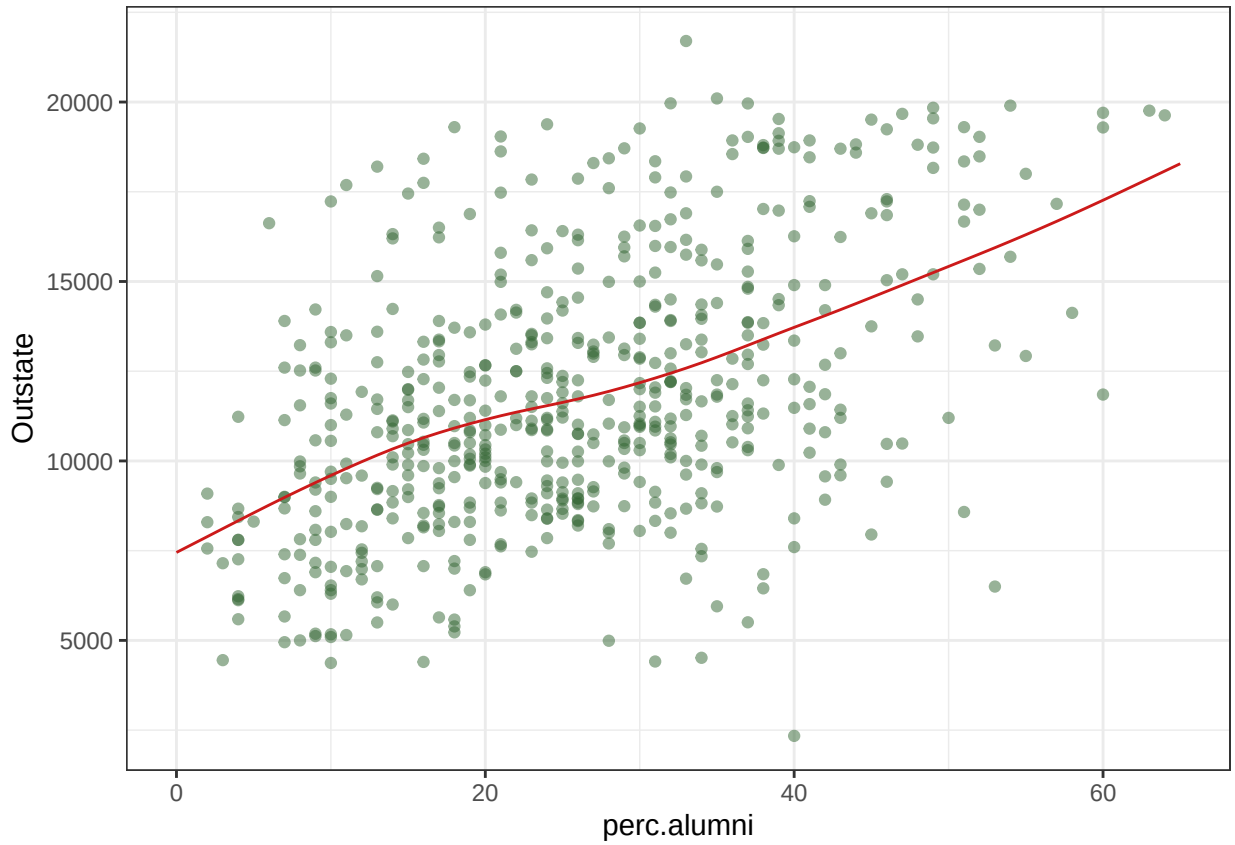
```
# Plot for df = 3
p +
  geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df_3,
            color = rgb(.8, .1, .1, 1)) + theme_bw()
```



The plot of smoothing spline fit for df = 3 is slightly curvy, where `perc.alomni` and `Outstate` have positive relationship. There is a curve around midpoint between 20 and 40 from x-axis, and everywhere else are almost linear.

Plot for df = 5

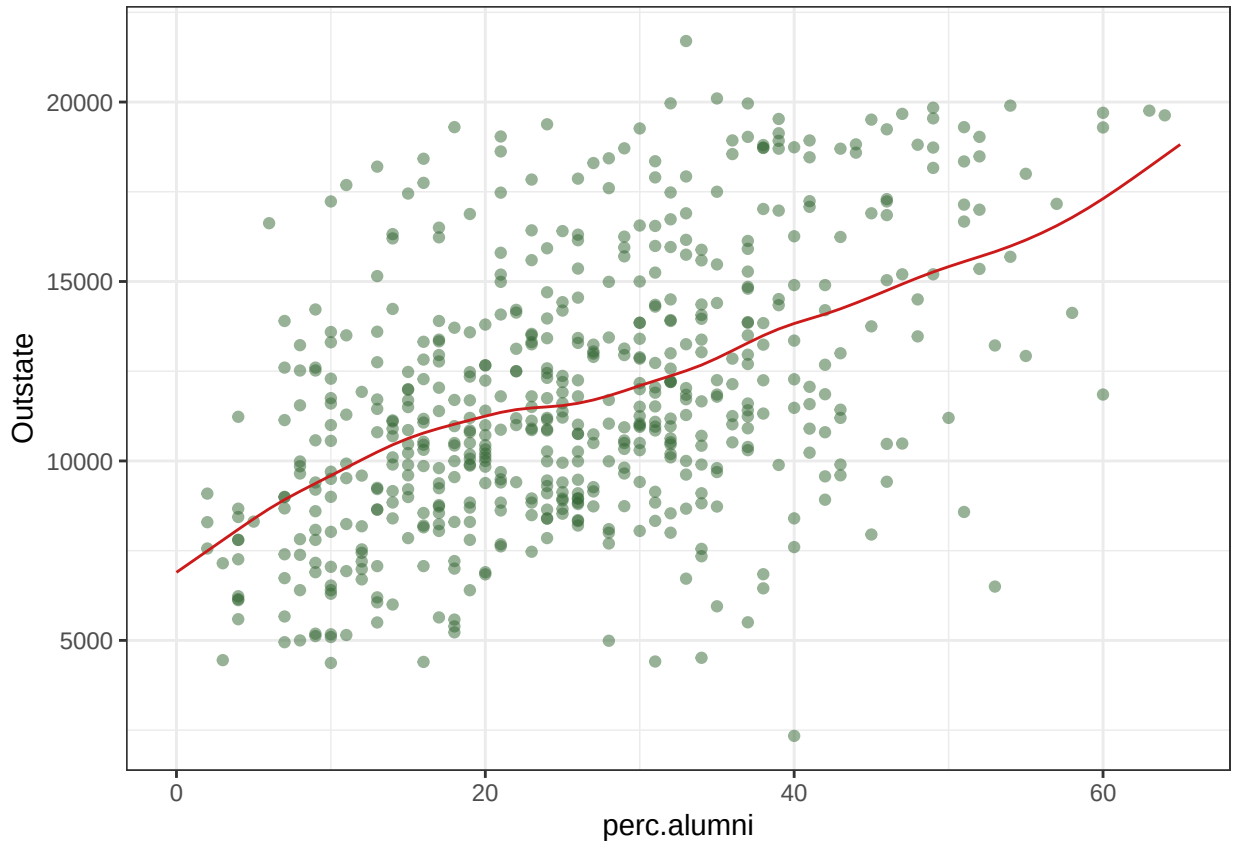
```
p +
  geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df_5,
            color = rgb(.8, .1, .1, 1)) + theme_bw()
```



The plot of the smoothing spline fit for df = 5 has a bit of a curve in the first half of the line, making it non-linear. There is a positive relationship between `perc.alomni` and `Outstate`. The plot is non-linear because the specified df is 5, which greater than 2 and makes the plot non-linear and slightly curvy.

Plot for df = 8

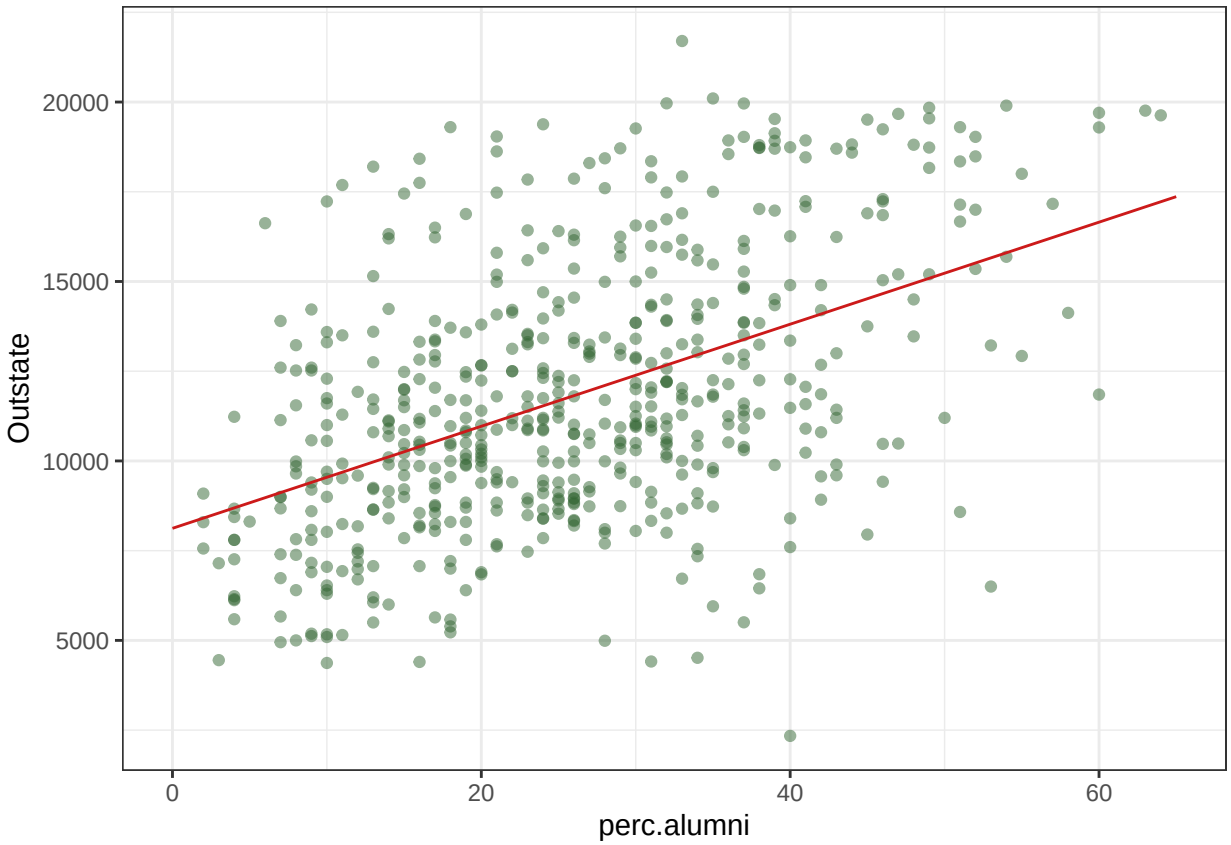
```
p +
  geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df_8,
            color = rgb(.8, .1, .1, 1)) + theme_bw()
```



The plot of the smoothing spline fit for df = 8 is the most non-linear since it has the highest df from above models. There is a positive relationship between `perc.alomni` and `Outstate`, with a larger curve in the first half of the line (near the lower values of `perc.alomni` and `Outstate`). The plot is non-linear because the specified df is 8, which quite a bit larger than 2 and makes the plot non-linear and curvy.

Plot for df obtained by cv

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



The plot of smoothing spline fit the df obtained by cv is the most linear, with a positive relationship between `perc.alomni` and `Outstate`. The plot is linear because the specified df is close to 2, making it similar to a second degree polynomial resulting in a linear plot.

Thus, we can see that when the degrees of freedom is small, the fitted line is close to linear, and it gets more and more wiggly as degrees of freedom increase.

(b)

Fit a generalized additive model (GAM) using all the predictors. Does your GAM model include all the predictors? Plot the results and explain your findings. Report the test error.

```
set.seed(123)

# fit GAM using all predictors
gam.fit_all <- train(x, y,
                     method = "gam",
                     trControl = ctrl1,
                     control = gam.control(maxit = 200))
gam.fit_all$bestTune
```

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

```
gam.fit_all$finalModel
```

```
##
## Family: gaussian
```

```
## Link function: identity
##
## Formula:
## .outcome ~ s(perc.alumni) + s(Terminal) + s(Books) + s(PhD) +
##     s(Grad.Rate) + s(Top10perc) + 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:
## 2.46 4.80 2.13 6.18 1.00 1.00 1.37
## 3.46 1.00 1.00 1.00 2.17 2.49 5.24
## 1.00 6.19  total = 43.48
##
## GCV score: 2834679
```

There are 13 predictors in the final GAM model obtained from selection of all predictors with GCV score 2598870, where the 2nd predictor `Terminal`, 7th predictor `Top25perc`, and 10th predictor `P.Undergrad` are zero.

```
# fit GAM using selection specification
gam.fit_select <- train(x, y, # test dataset
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE)),
                 trControl = ctrl1, # 10-fold CV
                 control = gam.control(maxit = 200))  # Adjusted due to failure to converge at default
gam.fit_select$bestTune
```

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

```
gam.fit_select$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc.alumni) + s(Terminal) + s(Books) + s(PhD) +
##     s(Grad.Rate) + s(Top10perc) + 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:
## 2.760 0.384 6.116 0.185 3.276 6.828 0.000
## 2.901 5.326 0.000 1.837 6.711 4.295 5.234
## 3.693 5.341  total = 55.89
##
## GCV score: 2822251
```

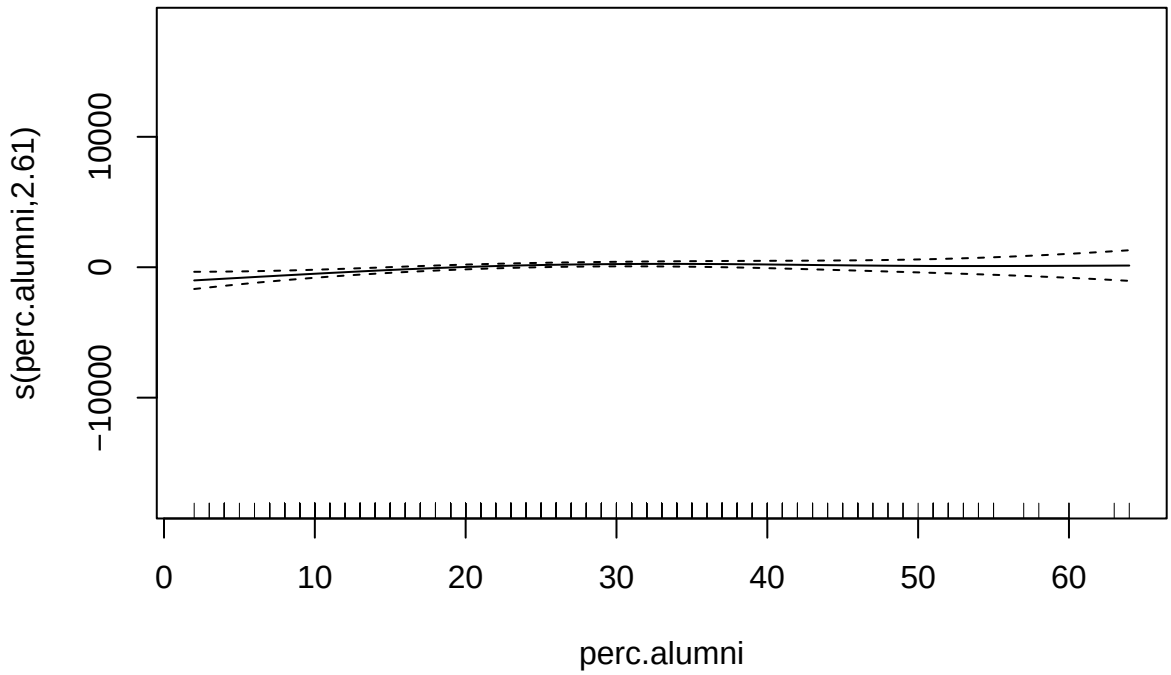
Same results as before. There are 13 predictors in the final GAM model obtained from selection of all predictors with GCV score 2598870, where the 2nd predictor `Terminal`, 7th predictor `Top25perc`, and 10th predictor `P.Undergrad` are zero. In conclusion, the full model contains all 16 predictors, while the selection specification model has 13 predictors, where the 2nd predictor `Terminal`, 7th predictor `Top25perc`, and 10th predictor `P.Undergrad` are removed from the final model.

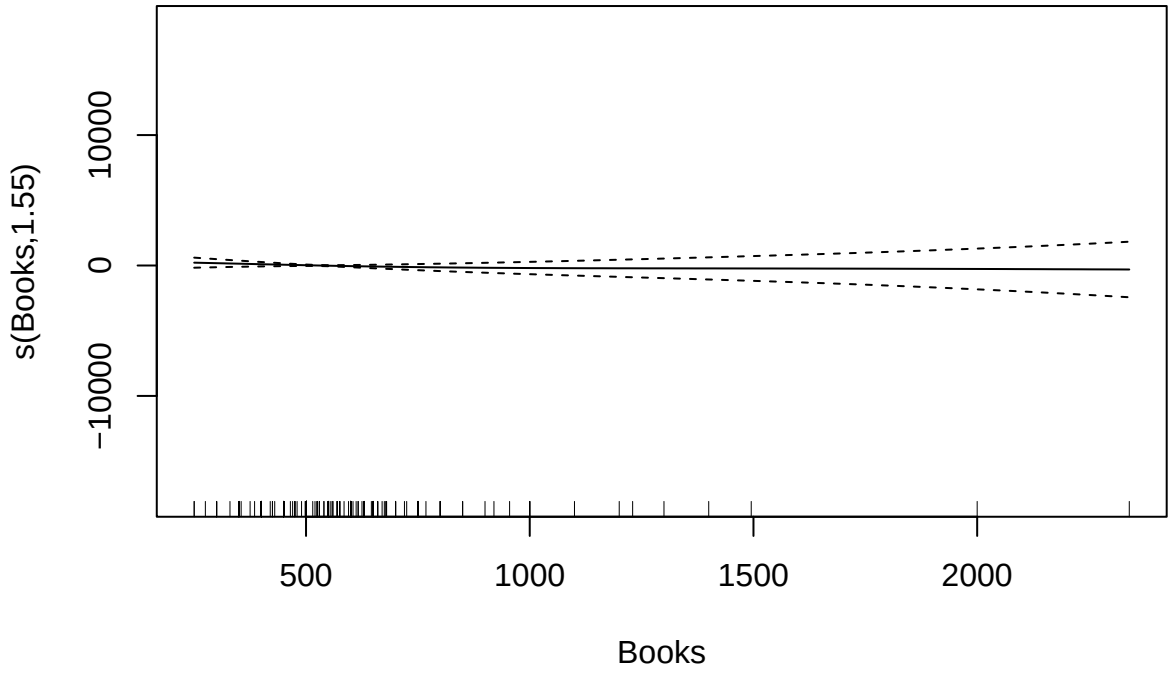
Plot

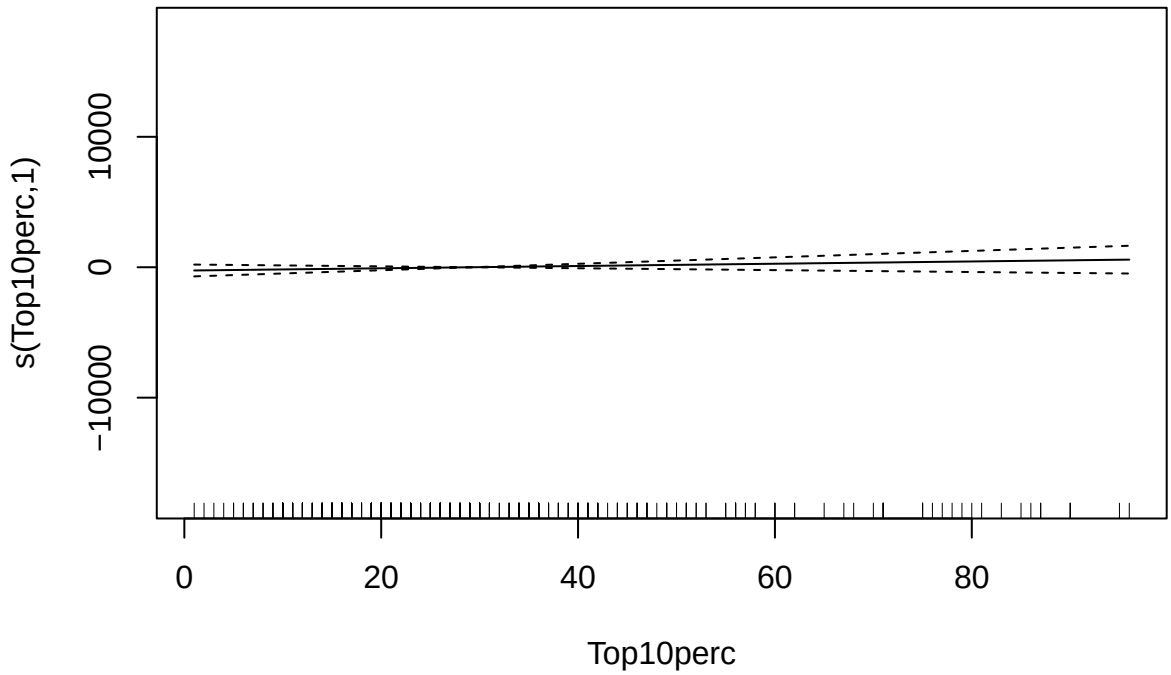
```
# Formula based on final GAM model with 13 predictors (gam.fit_select)
gam.m1 <- gam(Outstate ~ s(perc.alumni) + s(Books) + s(Top10perc) +
                s(Grad.Rate) + s(PhD) + s(S.F.Ratio) + s(Personal) +
                s(Room.Board) + s(Enroll) + s(Accept) + s(Apps) +
                s(F.Undergrad) + s(Expend),
              data = dat_train) # training dataset
summary(gam.m1)
```

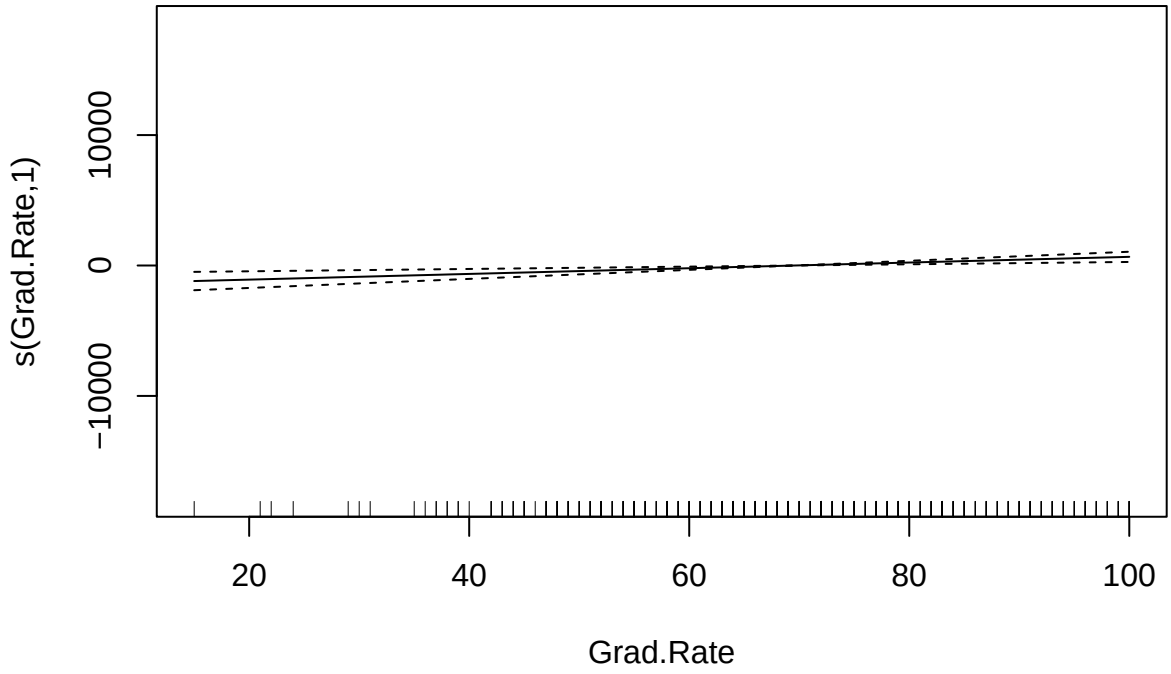
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Outstate ~ s(perc.alumni) + s(Books) + s(Top10perc) + s(Grad.Rate) +
##     s(PhD) + s(S.F.Ratio) + s(Personal) + s(Room.Board) + s(Enroll) +
##     s(Accept) + s(Apps) + s(F.Undergrad) + s(Expend)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11845.69      76.18   155.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df      F  p-value
## s(perc.alumni) 2.610  3.297  3.860 0.008326 **
## s(Books)       1.552  1.914  0.492 0.539398
## s(Top10perc)   1.000  1.000  1.198 0.274361
## s(Grad.Rate)   1.000  1.000 11.496 0.000763 ***
## s(PhD)         1.471  1.818  1.800 0.139907
## s(S.F.Ratio)   3.032  3.879  1.241 0.311459
## s(Personal)    2.808  3.526  3.078 0.023702 *
## s(Room.Board)  2.570  3.275 14.946  < 2e-16 ***
## s(Enroll)      1.000  1.000 16.304 6.47e-05 ***
## s(Accept)      2.713  3.466  5.558 0.000599 ***
## s(Apps)        1.000  1.000  5.595 0.018464 *
## s(F.Undergrad) 5.333  6.360  2.527 0.019288 *
## s(Expend)      6.037  7.182 21.611  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.803   Deviance explained = 81.7%
## GCV = 2.836e+06  Scale est. = 2.6286e+06  n = 453
```

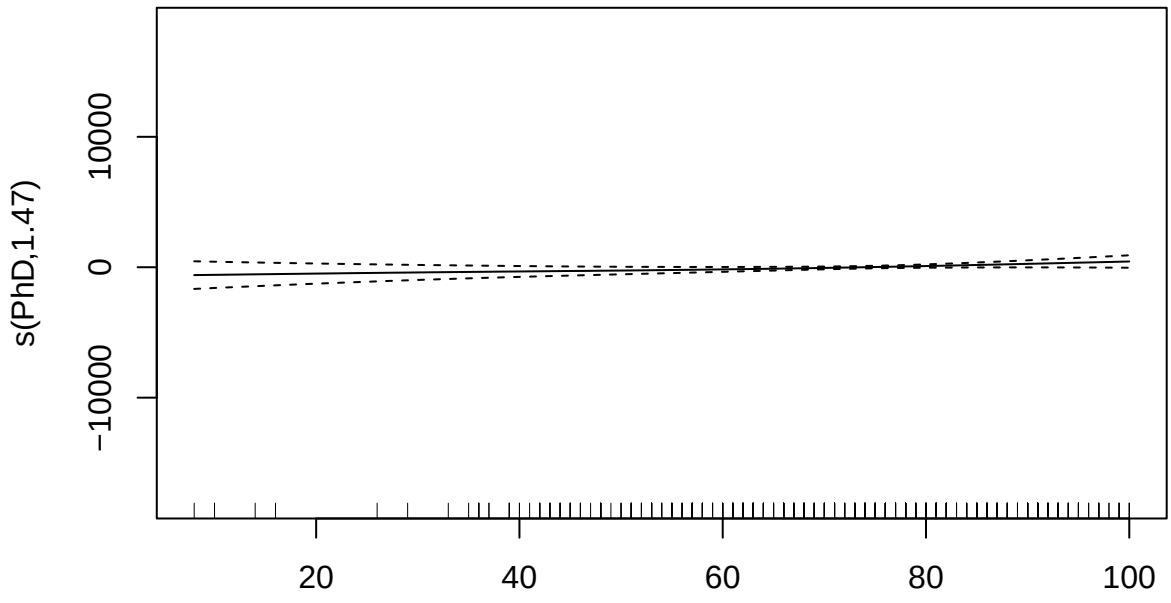
```
plot(gam.m1)
```

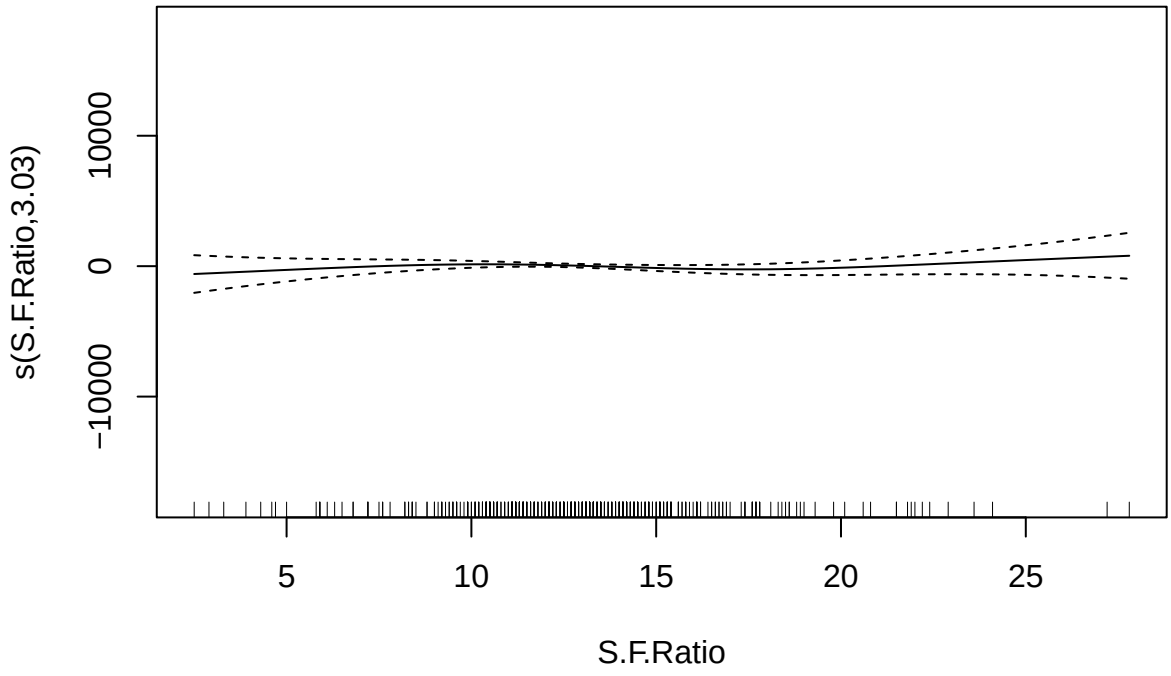

s(perc.alumni,2.61)
10000
0
−10000
0
10
20
30
40
50
60
perc.alumni

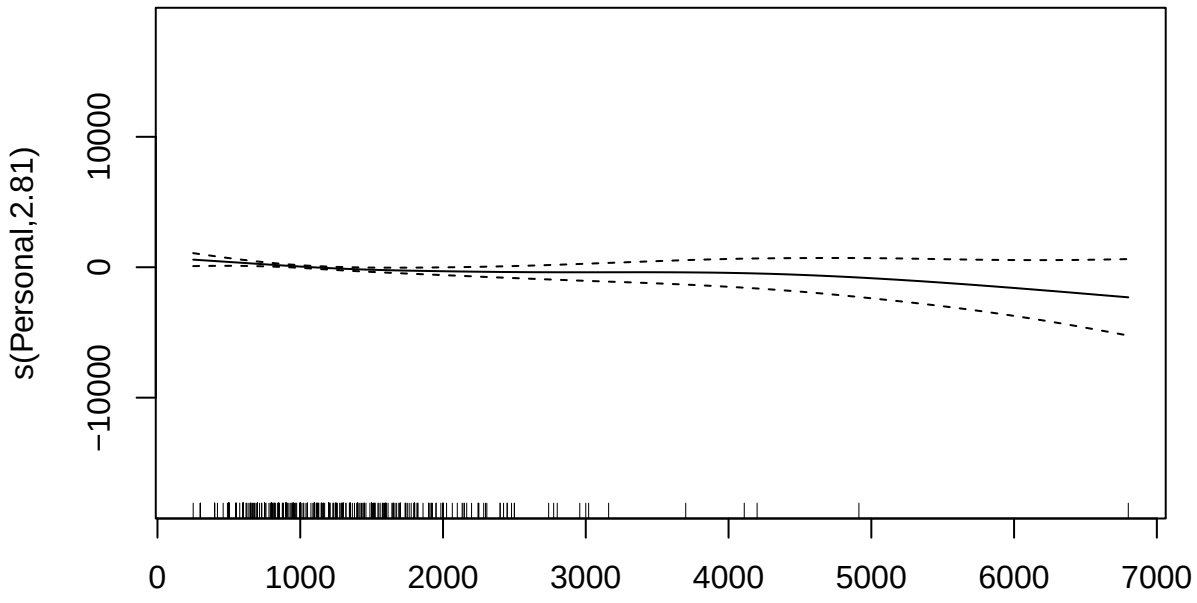

s(Books,1.55)
10000
0
−10000
500
1000
1500
2000
Books

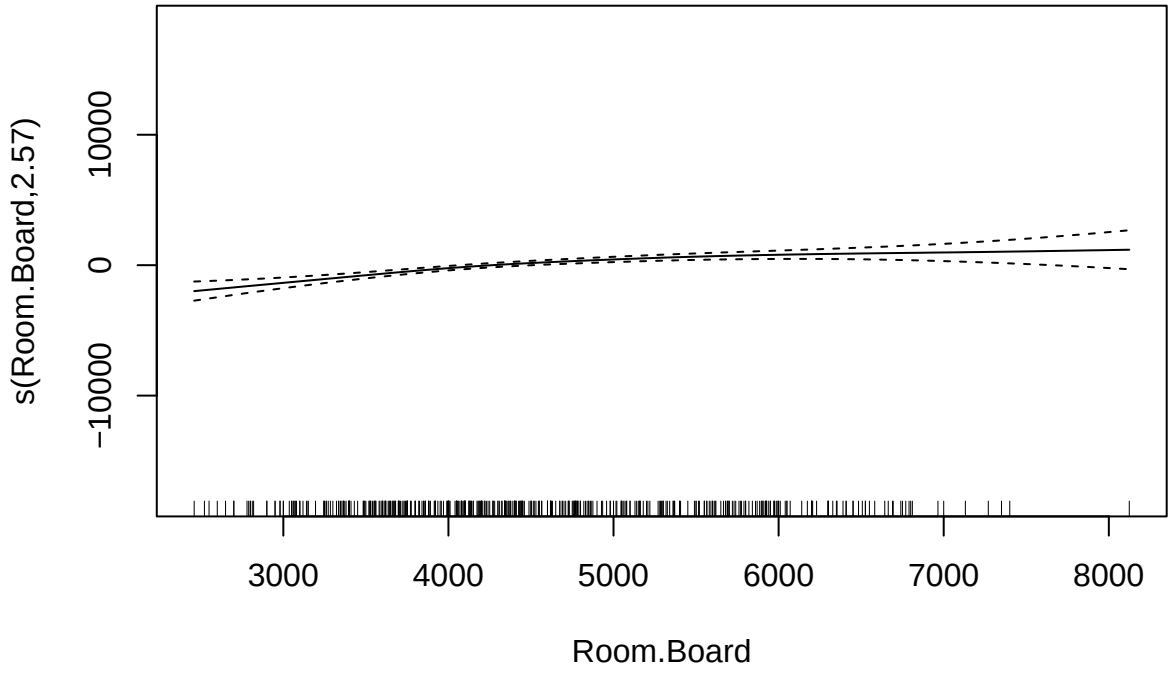

s(Top10perc,1)
10000
0
−10000
0
20
40
60
80
Top10perc

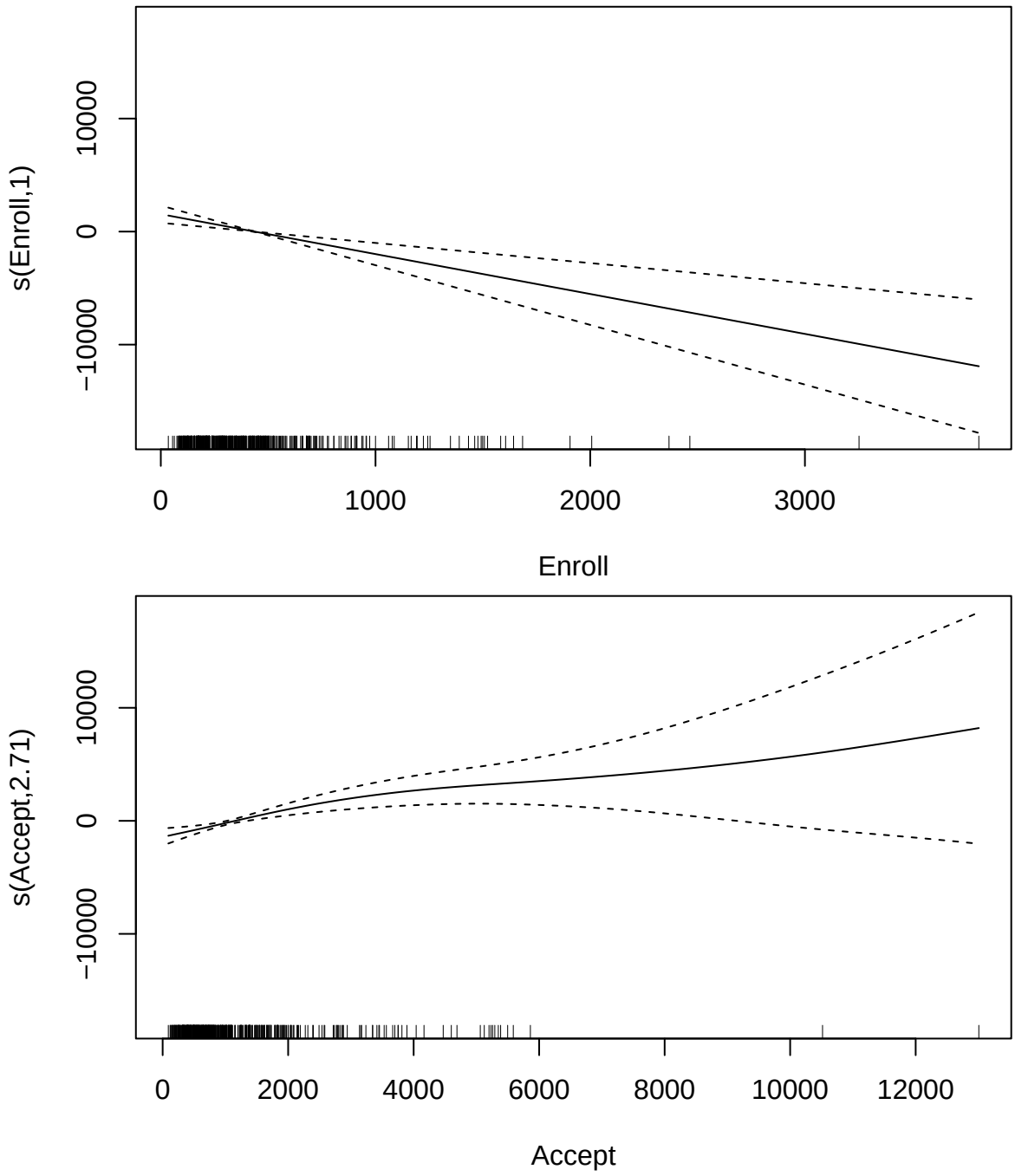

s(Grad.Rate,1)
10000
0
−10000
20
40
60
80
100
Grad.Rate

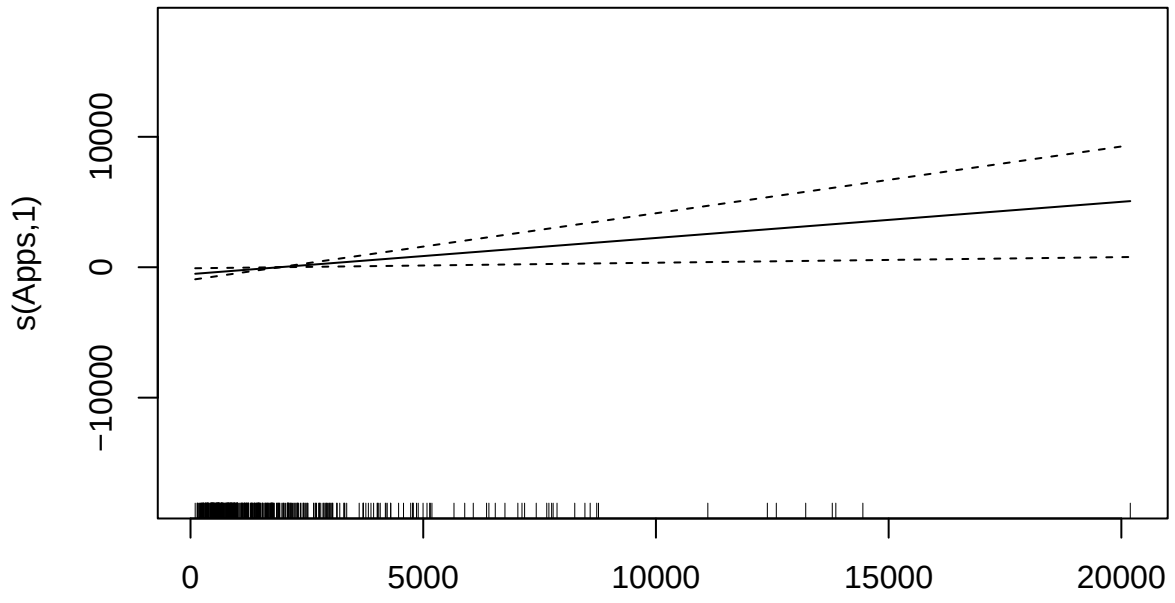

s(PhD,1.47)
10000
0
−10000
20
40
60
80
100
PhD

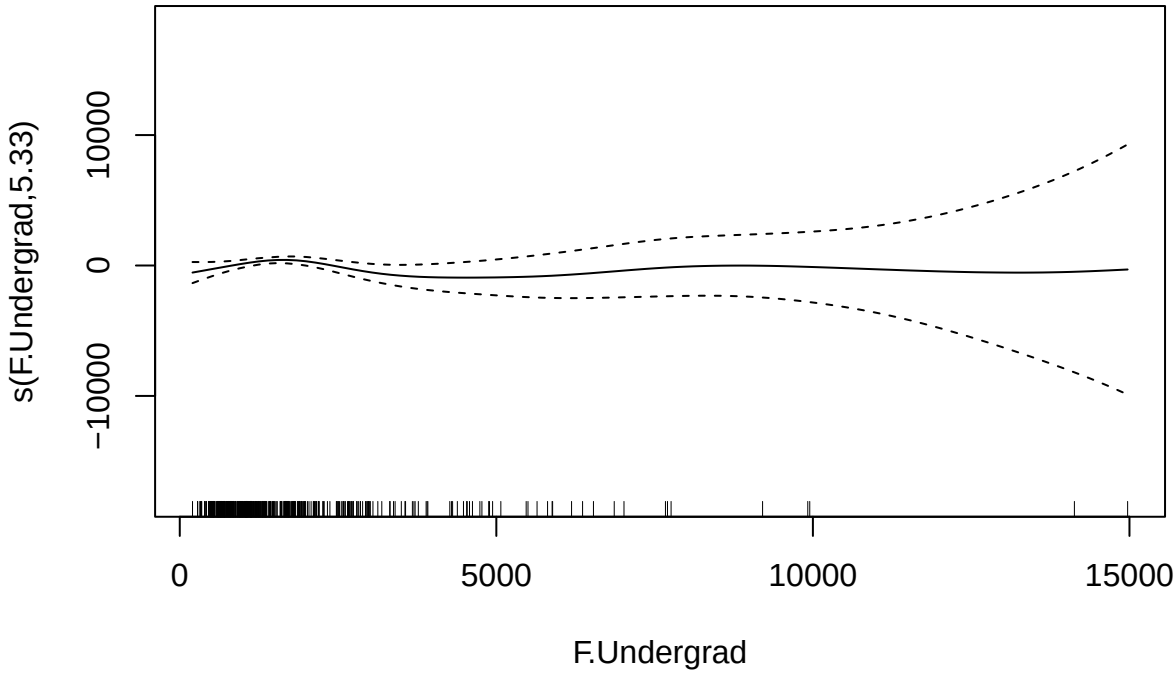

s(S.F.Ratio,3.03)
10000
0
−10000
5
10
15
20
25
S.F.Ratio

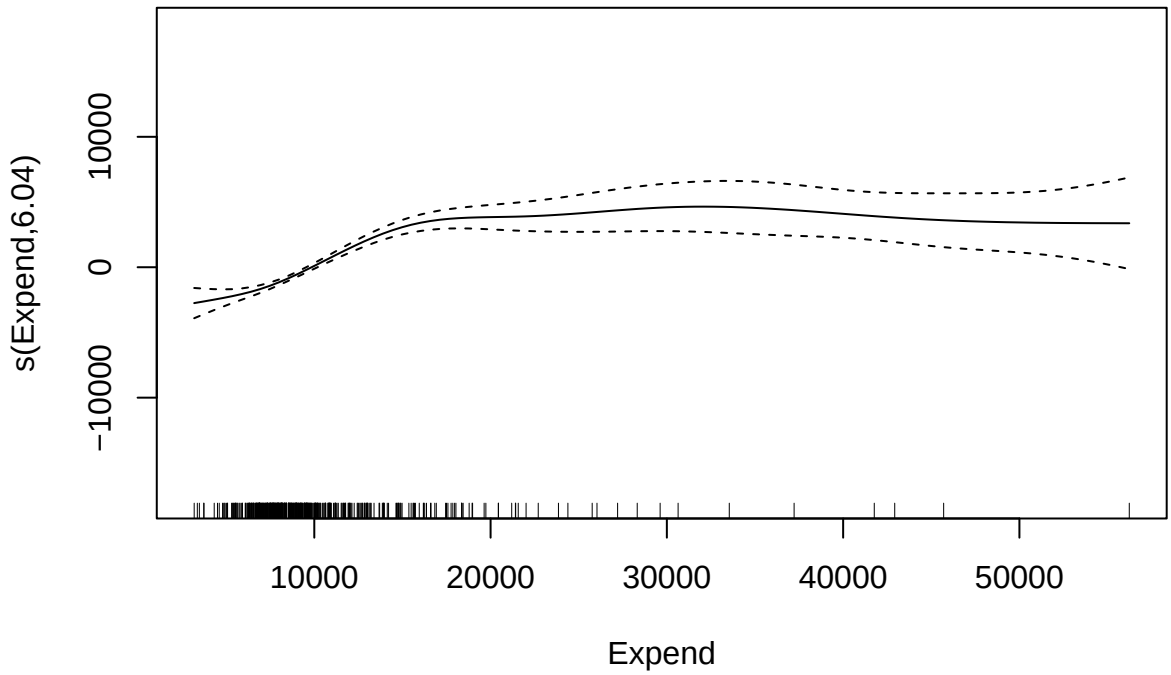

s(Personal,2.81)
10000
0
−10000
0
1000
2000
3000
4000
5000
6000
7000
Personal


s(Room.Board,2.57)
10000
0
−10000
3000
4000
5000
6000
7000
8000
Room.Board

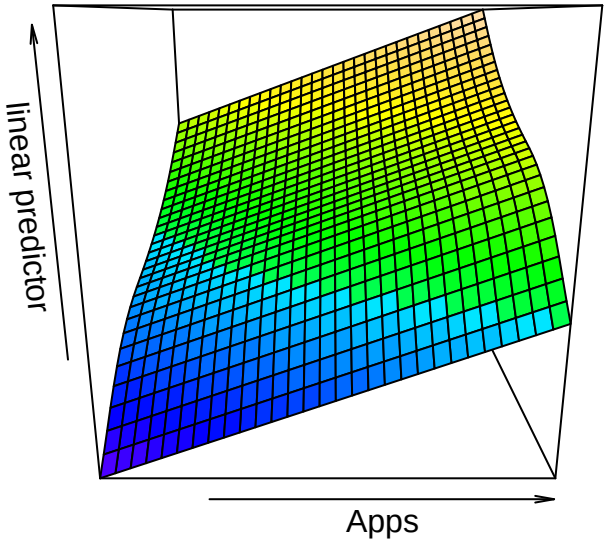

s(Enroll,1)
10000
0
−10000
0
1000
2000
3000
Enroll
s(Accept,2.71)
10000
0
−10000
0
2000
4000
6000
8000
10000
12000
Accept


s(Apps,1)
10000
0
−10000
0
5000
10000
15000
20000
Apps


s(F.Undergrad,5.33)
10000
0
−10000
0
5000
10000
15000
F.Undergrad




```
vis.gam(gam.m1, view = c("Apps", "Accept"),
        color = "topo")
```



According to the p value from summary table, some predictors may not be significant in the GAM model, including Books, Top10perc, and Personal. While look at the patterns at significant predictors, the plots are slightly different from summary table. Based on the plots, the predictors `perc.alumni`, `Grad.Rate`, `PhD`, `S.F.Ratio`, `Room.Board`, `Accept`, `Apps`, `F.Undergrad`, and `Expend` have positive relation with the outcome. The remaining predictors tend to have a negative or close to constant relationship with the outcome. The deviance explained by the model is 83.4%, adjusted R-squared value is 0.819, which is quite close to 1. Thus the GAM model fits the data quite well.

Test Error

```
set.seed(123)
```

```
gam.pred <- predict(gam.m1, newdata = x2)

test_error_gam <- mean((gam.pred - y2)^2)
test_error_gam
```

```
## [1] 2973833
```

```
RMSE_gam <- sqrt(test_error_gam)
RMSE_gam
```

```
## [1] 1724.48
```

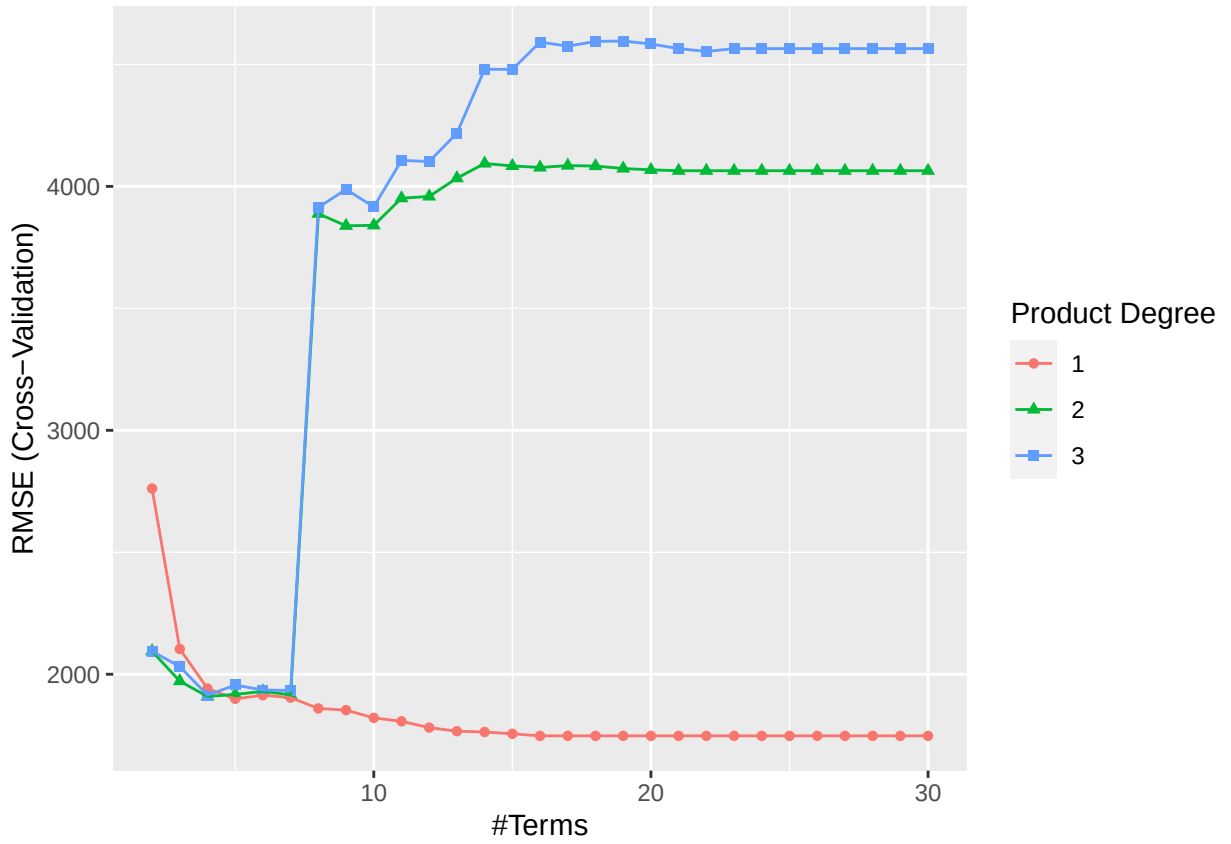
The test error is 5125770, and RMSE is 2264.016.

(c)

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. Report the test error.

Final model

```
set.seed(123)

# # create grid of all possible pairs that can take degree and nprune values
# mars_grid <- expand.grid(degree = 1:3, # number of possible product hinge functions in 1 term
#                          nprune = 2:16) # Upper bound of number of terms in model
#
# mars.fit <- train(x, y, # training dataset
#                   method = "earth",
#                   tuneGrid = mars_grid,
#                   trControl = ctrl1) # 10-fold CV
#
# ggplot(mars.fit)
# mars.fit$bestTune
# coef(mars.fit$finalModel)

# create wider range of nprune to include the minimum RMSE obtained from CV
mars_grid <- expand.grid(degree = 1:3, # number of possible product hinge functions in 1 term
                         nprune = 2:30) # Upper bound of number of terms in model

mars.fit <- train(x, y, # training dataset
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1) # 10-fold CV

ggplot(mars.fit)
```




```
mars.fit$bestTune
```

```
##    nprune degree
## 15     16      1
```

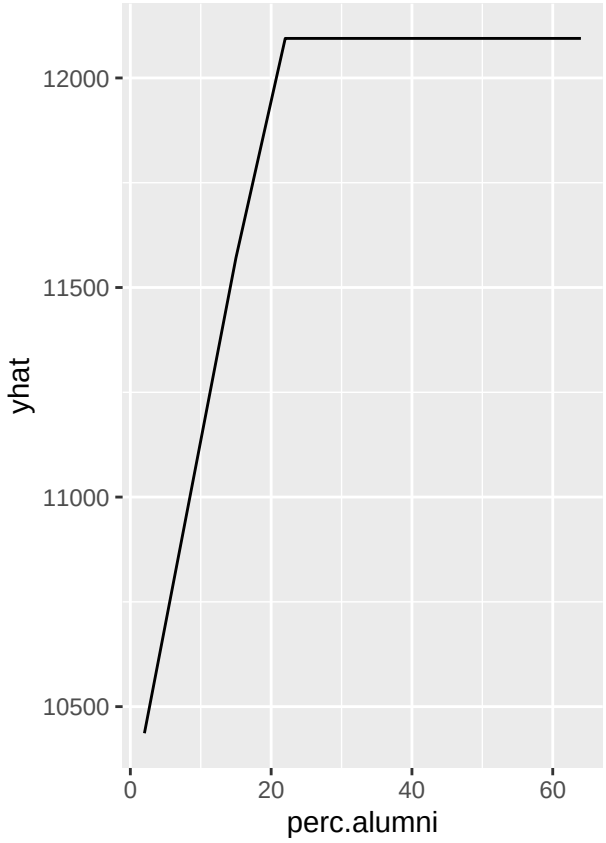
```
coef(mars.fit$finalModel)
```

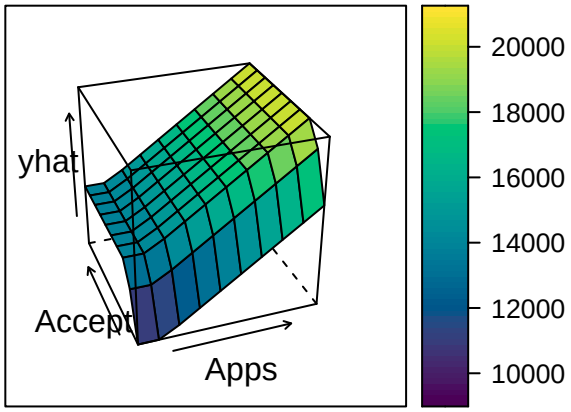
```
##          (Intercept)     h(Expend-15622)  h(Room.Board-4460)  h(4460-Room.Board)
##        10684.3159852          -0.7227653           0.3113264          -1.1274658
##     h(79-Grad.Rate)    h(1300-Personal) h(F.Undergrad-1350) h(1350-F.Undergrad)
##         -28.9480175           1.0471977          -0.4456624          -1.2719896
##        h(Apps-2694)   h(21-perc.alumni)      h(Expend-6898)       h(862-Enroll)
##           0.3774909         -87.2568633           0.7187916           4.9263485
##      h(2165-Accept)
##          -2.0063276
```

Partial dependence plot

```
p1 <- pdp::partial(mars.fit,
                   pred.var = c("perc.alumni"),
                   grid.resolution = 10) %>%
  autoplot()

# Plot of an interaction partial dependence plot between arbitrary predictors in the final model Apps a
p2 <- pdp::partial(mars.fit,
                   pred.var = c("Apps", "Accept"),
                   grid.resolution = 10) %>%
  pdp::plotPartial(levelplot = FALSE,
```

```
                  zlab = "yhat",
                  drape = TRUE,
                  screen = list(z = 20, x = -60))

# combine two plots
grid.arrange(p1, p2, ncol = 2)
```





Test error

```
set.seed(123)

mars.pred <- predict(mars.fit, newdata = x2)

test_error_mars <- mean((mars.pred - y2)^2)
test_error_mars
```

```
## [1] 2979788
```

```
RMSE_mars <- sqrt(test_error_mars)
RMSE_mars
```

```
## [1] 1726.206
```

The MARS test error is 3483157, and RMSE is 1866.322. Since MARS has smaller test error than GAM model, it is better than GAM model.

(d)

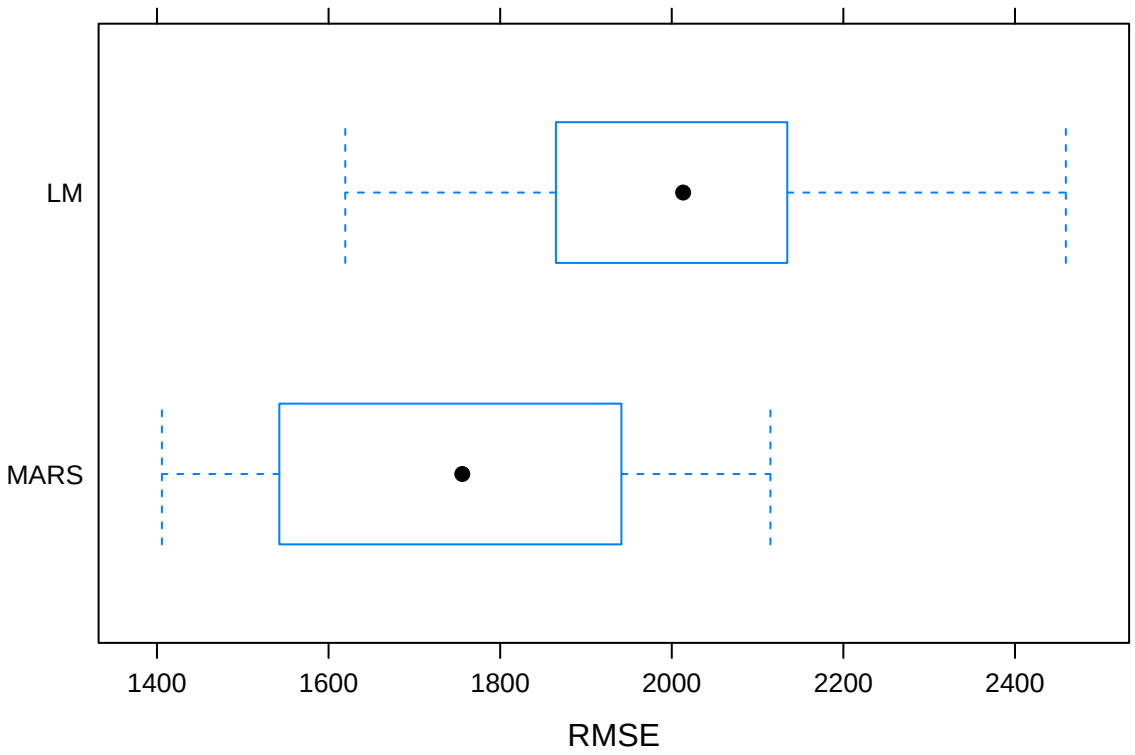
In this data example, do you prefer the use of MARS model over a linear model when predicting the out-of-state tuition? Why? For general applications, do you think MARS is a better approach compared to a linear model?

MARS model over a linear model

```
set.seed(123)
model.lm <- train(x, y,
                  method = "lm",
                  trControl = ctrl1)

resamp <- resamples(list(MARS = mars.fit, LM = model.lm))

summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: MARS, LM
## Number of resamples: 10
##
## MAE
##          Min.  1st Qu.  Median     Mean  3rd Qu.     Max. NA's
## MARS 1101.304 1185.060 1375.65 1340.834 1465.270 1556.097    0
## LM   1381.753 1472.153 1541.55 1578.644 1674.026 1828.710    0
##
## RMSE
##          Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## MARS 1405.699 1571.878 1755.857 1747.490 1916.005 2114.953    0
## LM   1619.446 1872.979 2013.260 1999.312 2127.698 2459.287    0
##
## Rsquared
##           Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## MARS 0.7017110 0.7506671 0.7855974 0.7785106 0.8170472 0.8391766    0
## LM   0.5771815 0.7212143 0.7228899 0.7102796 0.7407168 0.7567040    0
```

```
bwplot(resamp, metric = "RMSE")
```



I prefer the MARS model over a linear model. The best model for predicting the out-of-state tuition is the MARS model since it has the lowest mean value of RMSE comparing to a linear model.

Compare MARS and linear models for general applications [10pts/100pts]

For general applications, which is better always depends on the underlying true model, so neither model will always be better.