# 8106hw2

Ze Li

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

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

```
college=read.csv("/Users/zeze/Library/Mobile Documents/com~apple~CloudDocs/2024/24S BIST P8106 DS II/hw
indexTrain <- createDataPartition(y = college$Outstate, p = 0.8, list = FALSE)
train <- college[indexTrain, ]
test <- college[-indexTrain, ]
head(train)
```

```
##                        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
```

```
# matrix of predictors
x_train <- model.matrix(Outstate ~ . - College, train)[, -1]
head(x_train)
```

```
##   Apps Accept Enroll Top10perc Top25perc F.Undergrad P.Undergrad Room.Board
## 1 1660   1232    721        23        52        2885         537       3300
## 2 2186   1924    512        16        29        2683        1227       6450
## 3 1428   1097    336        22        50        1036          99       3750
## 4  417    349    137        60        89         510          63       5450
## 5  193    146     55        16        44         249         869       4120
## 6  587    479    158        38        62         678          41       3335
##   Books Personal PhD Terminal S.F.Ratio perc.alumni Expend Grad.Rate
## 1   450     2200  70       78      18.1          12   7041        60
## 2   750     1500  29       30      12.2          16  10527        56
## 3   400     1165  53       66      12.9          30   8735        54
## 4   450      875  92       97       7.7          37  19016        59
## 5   800     1500  76       72      11.9           2  10922        15
## 6   500      675  67       73       9.4          11   9727        55
```

```r
# vector of response
y_train <- train$Outstate
# matrix of predictors
x_test <- model.matrix(Outstate ~ . - College, test)[, -1]
# vector of response
y_test <- test$Outstate
```

## Smoothing spline

(a) Fit smoothing spline models to predict out-of-state tuition (Outstate) using the percentage of alumni who donate (perc.alumni) as the only predictor, across a range of degrees of freedom. Plot the model fits for each degree of freedom. Describe the observed patterns that emerge with varying degrees of freedom. Select an appropriate degree of freedom for the model and plot this optimal fit. Explain the criteria you used to determine the best choice of degree of freedom.

## Polynomial regression

```r
fit1 <- lm(Outstate ~ perc.alumni, data = train)
fit2 <- lm(Outstate ~ poly(perc.alumni,2), data = train)
fit3 <- lm(Outstate ~ poly(perc.alumni,3), data = train)
fit4 <- lm(Outstate ~ poly(perc.alumni,4), data = train)
fit5 <- lm(Outstate ~ poly(perc.alumni,5), data = train)
anova(fit1, fit2, fit3, fit4, fit5)
```

```
## Analysis of Variance Table
##
## Model 1: Outstate ~ perc.alumni
## Model 2: Outstate ~ poly(perc.alumni, 2)
## Model 3: Outstate ~ poly(perc.alumni, 3)
## Model 4: Outstate ~ poly(perc.alumni, 4)
## Model 5: Outstate ~ poly(perc.alumni, 5)
##   Res.Df        RSS Df Sum of Sq      F  Pr(>F)
## 1    451 4891042648
## 2    450 4891041325  1      1323 0.0001 0.99120
## 3    449 4858075545  1  32965780 3.0369 0.08208 .
```

```
## 4    448 4853535503  1   4540043 0.4182 0.51815
## 5    447 4852208987  1   1326516 0.1222 0.72682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Use `anova()` to test the null hypothesis that a simpler model is sufficient to explain the data against the alternative hypothesis that a more complex model is required. In this case, we need a more complex model.

## smoothing.spline

```
perc.alumni.grid <- seq(from = -10, to = 110, by = 1)
fit.ss <- smooth.spline(train$perc.alumni, train$Outstate)
fit.ss$df
```

```
## [1] 2.000234
```

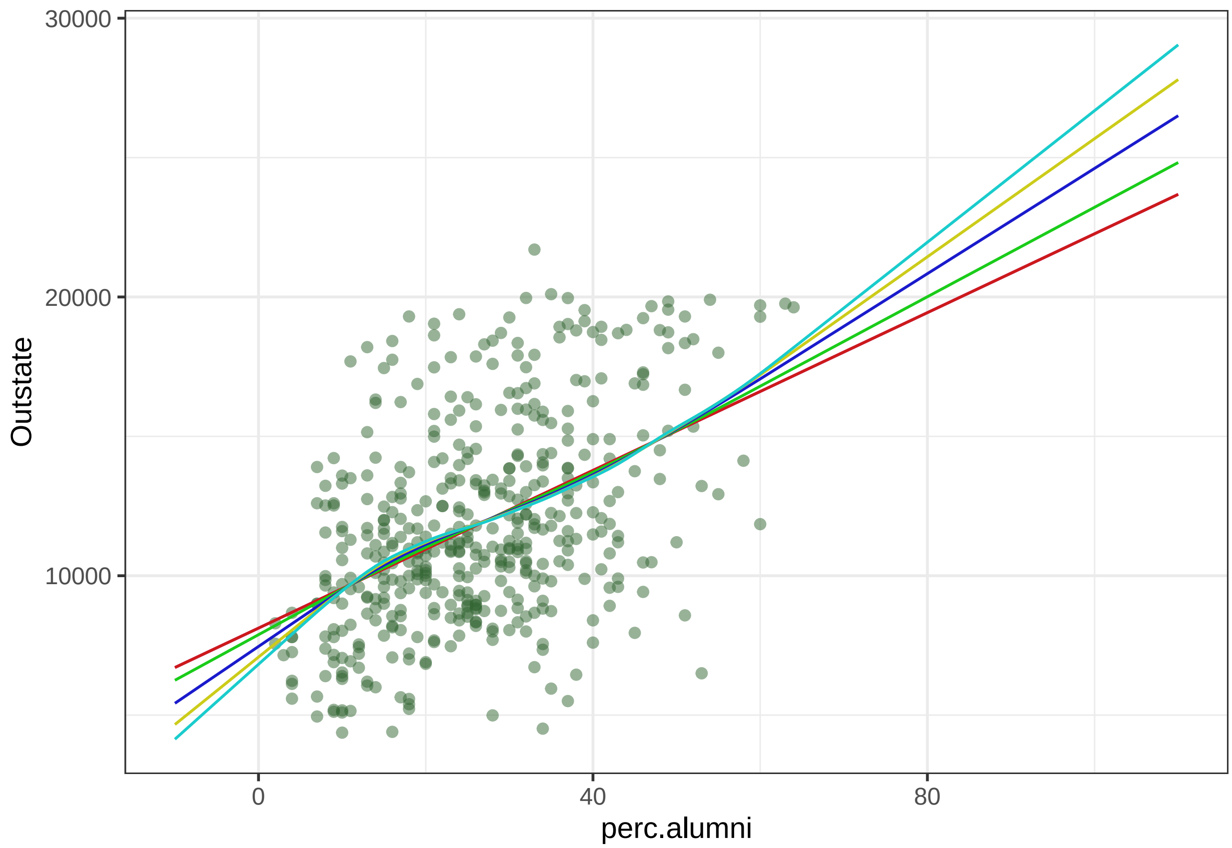
```
fit.ss2 <- smooth.spline(train$perc.alumni, train$Outstate,df=2)
fit.ss3 <- smooth.spline(train$perc.alumni, train$Outstate,df=3)
fit.ss4 <- smooth.spline(train$perc.alumni, train$Outstate,df=4)
fit.ss5 <- smooth.spline(train$perc.alumni, train$Outstate,df=5)
fit.ss6 <- smooth.spline(train$perc.alumni, train$Outstate,df=6)

pred.ss <- predict(fit.ss2,x = perc.alumni.grid)
pred.ss2 <- predict(fit.ss2,x = perc.alumni.grid)
pred.ss3 <- predict(fit.ss3,x = perc.alumni.grid)
pred.ss4 <- predict(fit.ss4,x = perc.alumni.grid)
pred.ss5 <- predict(fit.ss5,x = perc.alumni.grid)
pred.ss6 <- predict(fit.ss6,x = perc.alumni.grid)

pred.ss.df <- data.frame(pred = pred.ss2$y,perc.alumni = perc.alumni.grid)
pred.ss2.df <- data.frame(pred = pred.ss2$y,perc.alumni = perc.alumni.grid)
pred.ss3.df <- data.frame(pred = pred.ss3$y,perc.alumni = perc.alumni.grid)
pred.ss4.df <- data.frame(pred = pred.ss4$y,perc.alumni = perc.alumni.grid)
pred.ss5.df <- data.frame(pred = pred.ss5$y,perc.alumni = perc.alumni.grid)
pred.ss6.df <- data.frame(pred = pred.ss6$y,perc.alumni = perc.alumni.grid)

p <- ggplot(data = train, aes(x = perc.alumni, y = Outstate)) +
     geom_point(color = rgb(.2, .4, .2, .5))

p +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df,
          color = rgb(.8, .1, .8, 1)) + theme_bw() +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss2.df,
          color = rgb(.8, .1, .1, 1)) + theme_bw() +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss3.df,
          color = rgb(.1, .8, .1, 1)) + theme_bw() +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss4.df,
          color = rgb(.1, .1, .8, 1)) + theme_bw() +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss5.df,
          color = rgb(.8, .8, .1, 1)) + theme_bw() +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss6.df,
          color = rgb(.1, .8, .8, 1)) + theme_bw()
```




We can see that the model starts to follow the noise in the data rather than the underlying trend as the degree of freedom increases. The best degree of freedom should strike a balance between flexibility and smoothness; it fits the general trend of the data without overfitting. This is typically done using a criterion such as the AIC, BIC, or Cross-Validation for smoothing splines. In this case, the best fit degree of freedom is 2.0002343.

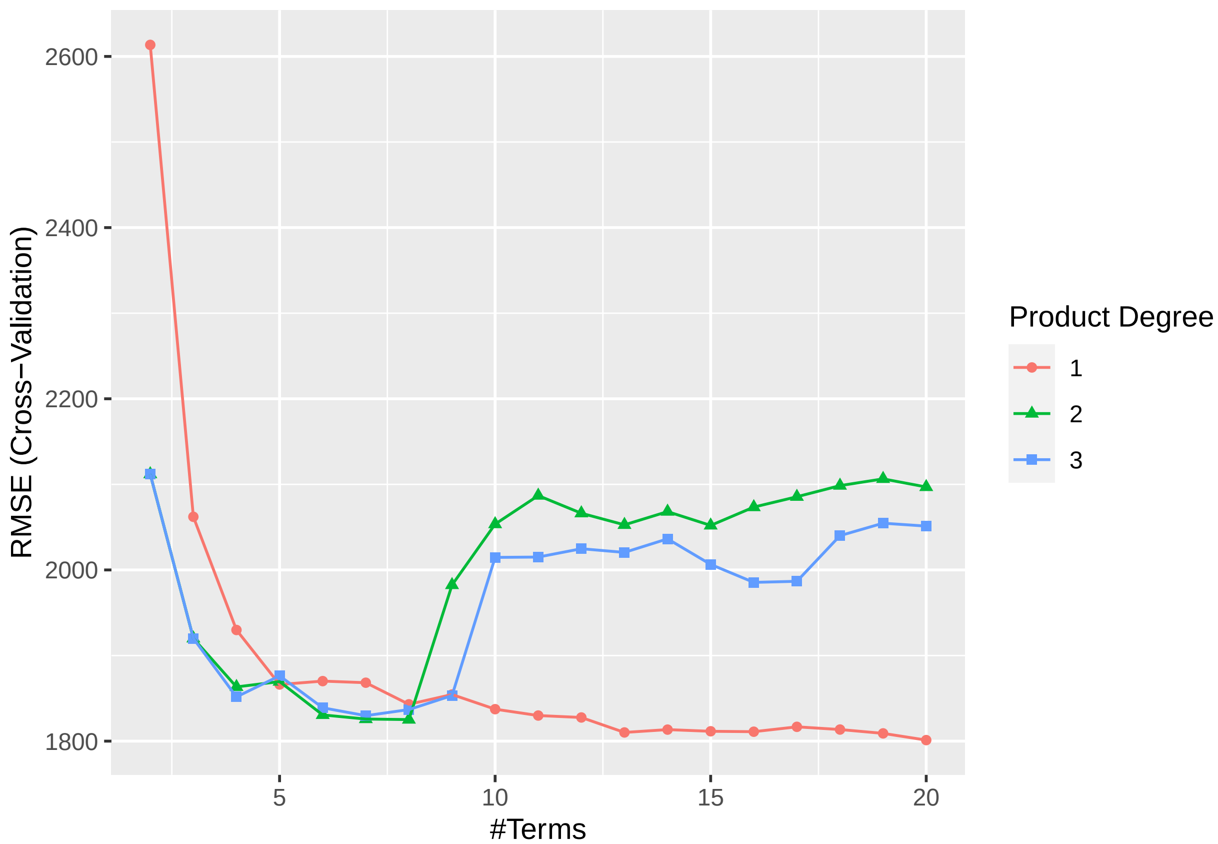
## MARS

(b) Train a multivariate adaptive regression spline (MARS) model to predict the response variable. Report the regression function. Present the partial dependence plot of an arbitrary predictor in your model. Report the test error.

```
ctrl1 <- trainControl(method = "cv", number = 10)
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:20)

set.seed(2)
mars.fit <- train(x_train, y_train,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)
```




```r
mars.fit$bestTune
```

```
##    nprune degree
## 19     20      1
```

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

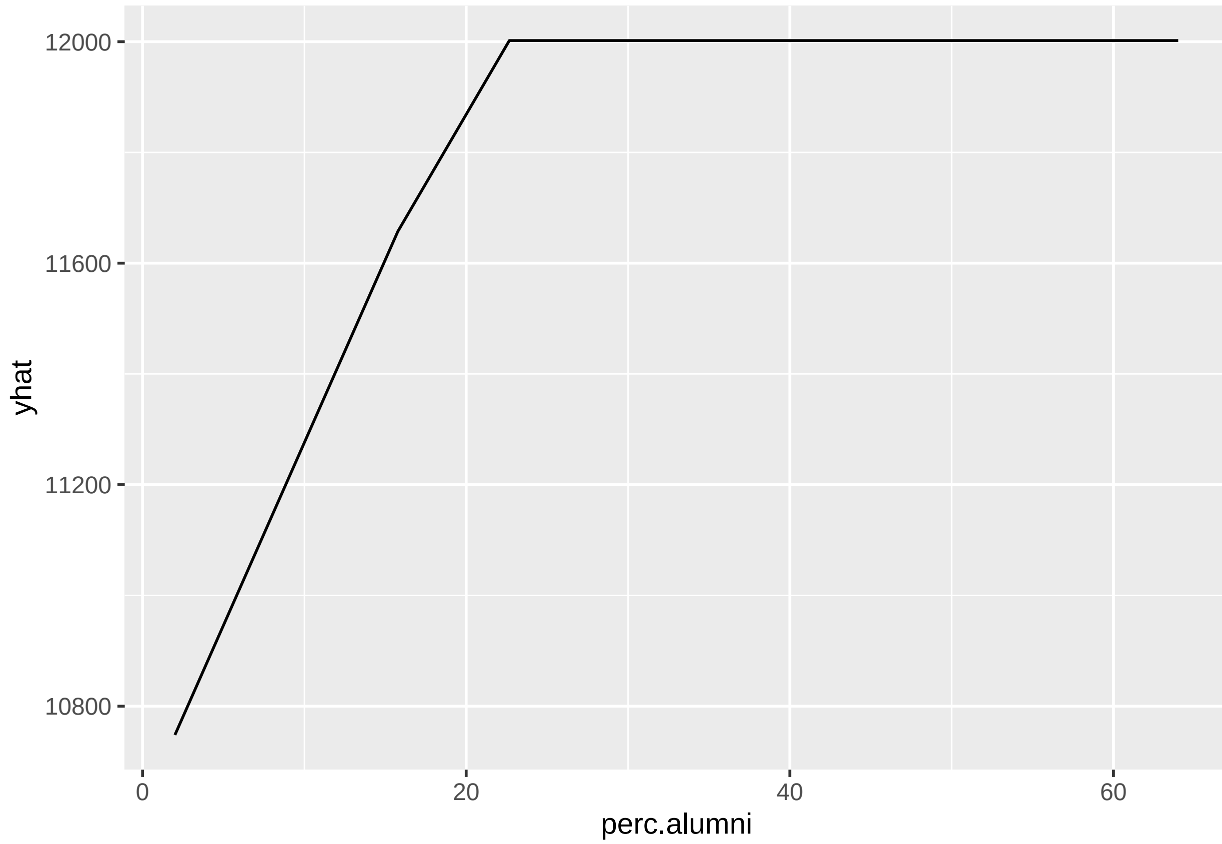
```
##          (Intercept)     h(Expend-15387)     h(83-Grad.Rate)  h(Room.Board-4100)
##         9734.0819348          -0.6563929         -30.3115498           0.3251654
##   h(4100-Room.Board)     h(Personal-900)     h(900-Personal) h(F.Undergrad-1410)
##           -1.1002740          -0.3124506           1.5499636          -0.3770000
## h(1410-F.Undergrad)        h(Apps-3708)   h(21-perc.alumni)            h(PhD-81)
##           -1.2396759           0.3445205         -65.9986988          68.6917486
##       h(Expend-4957)      h(2081-Accept)        h(820-Enroll)
##            0.6540199          -1.7080009           4.1125837
```

The regression function's coefficient is 9734.0819348, -0.6563929, -30.3115498, 0.3251654, -1.100274, -0.3124506, 1.5499636, -0.377, -1.2396759, 0.3445205, -65.9986988, 68.6917486, 0.6540199, -1.7080009, 4.1125837.

Therefore, Outstate = 13046.3266 - 0.5971 * h(15411-Expend) - 31.6804 * h(80-Grad.Rate) -1.0922 * h(4725-Room.Board) +1.1726 * h(1400-Personal) - -1.5732 * h(1263-F.Undergrad) + 0.4923 * h(Apps-1416) -30.1078 * h(51-perc.alumni) + 64.4511 * h(PhD-79) -1.7573 * h(Enroll-1462) + 3.8888 * h(1462-Enroll) -1.1475 * h(1557-Accept).

The partial dependence plot of an arbitrary predictor is

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




The test error is

```r
mars.pred <- predict(mars.fit, newdata = x_test)
sqrt(mean((y_test - mars.pred)^2))
```

```
## [1] 1689.95
```

## GAM

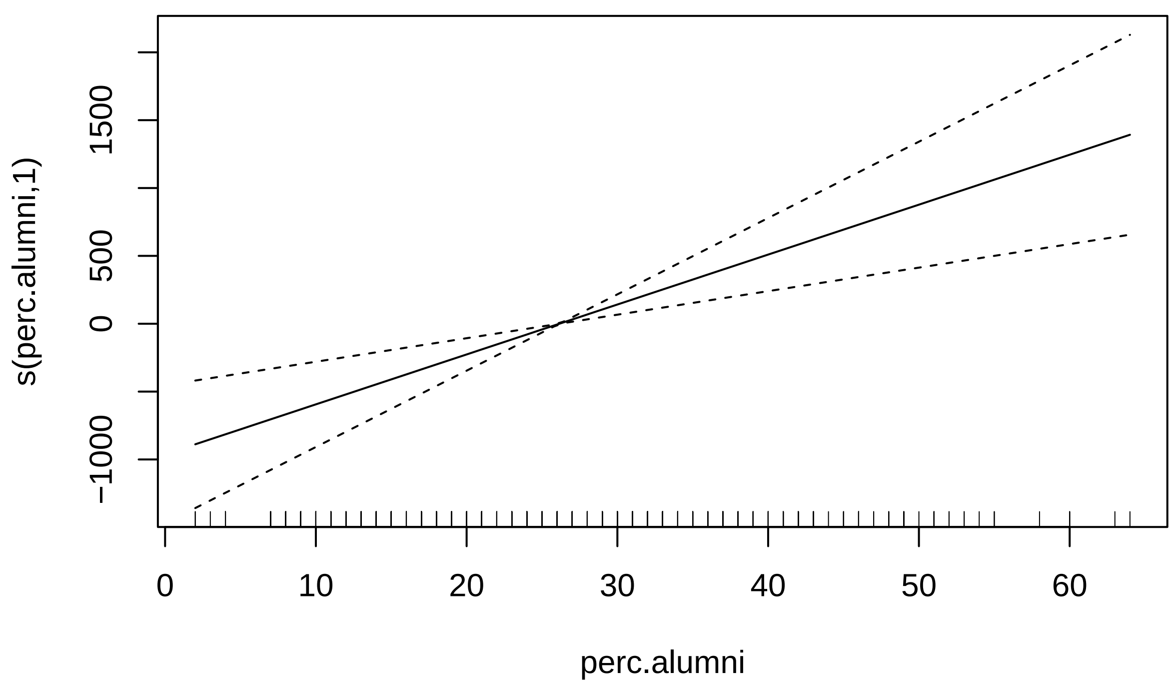
(c) Construct a generalized additive model (GAM) to predict the response variable. Does your GAM model include all the predictors? For the nonlinear terms included in your model, generate plots to visualize these relationships and discuss your observations. Report the test error.

```r
gam.m1 <- gam(Outstate ~ perc.alumni + Apps + Accept + Enroll
              + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Room.Board
              + Books + Personal + PhD + Terminal + S.F.Ratio + Expend + Grad.Rate,
              data = train)
gam.m2 <- gam(Outstate ~ s(perc.alumni) + Apps + Accept + Enroll
              + Top10perc + Top25perc + F.Undergrad + P.Undergrad + Room.Board
              + Books + Personal + PhD + Terminal + S.F.Ratio + Expend + Grad.Rate,
              data = train)
```

```r
anova(gam.m1, gam.m2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ perc.alumni + Apps + Accept + Enroll + Top10perc +
##     Top25perc + F.Undergrad + P.Undergrad + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + Expend + Grad.Rate
## Model 2: Outstate ~ s(perc.alumni) + Apps + Accept + Enroll + Top10perc +
##     Top25perc + F.Undergrad + P.Undergrad + Room.Board + Books +
##     Personal + PhD + Terminal + S.F.Ratio + Expend + Grad.Rate
##   Resid. Df Resid. Dev         Df   Deviance      F    Pr(>F)
## 1       436 1692874550
## 2       436 1692874550 8.9801e-10 0.00026584 0.0762 1.056e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(gam.m2)
```




```r
ctrl1 <- trainControl(method = "cv", number = 10)

set.seed(2)
gam.fit <- train(x_train, y_train,
                 method = "gam",
```

```r
                 tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
                 trControl = ctrl1)
```

```
## Warning: model fit failed for Fold08: 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.
```

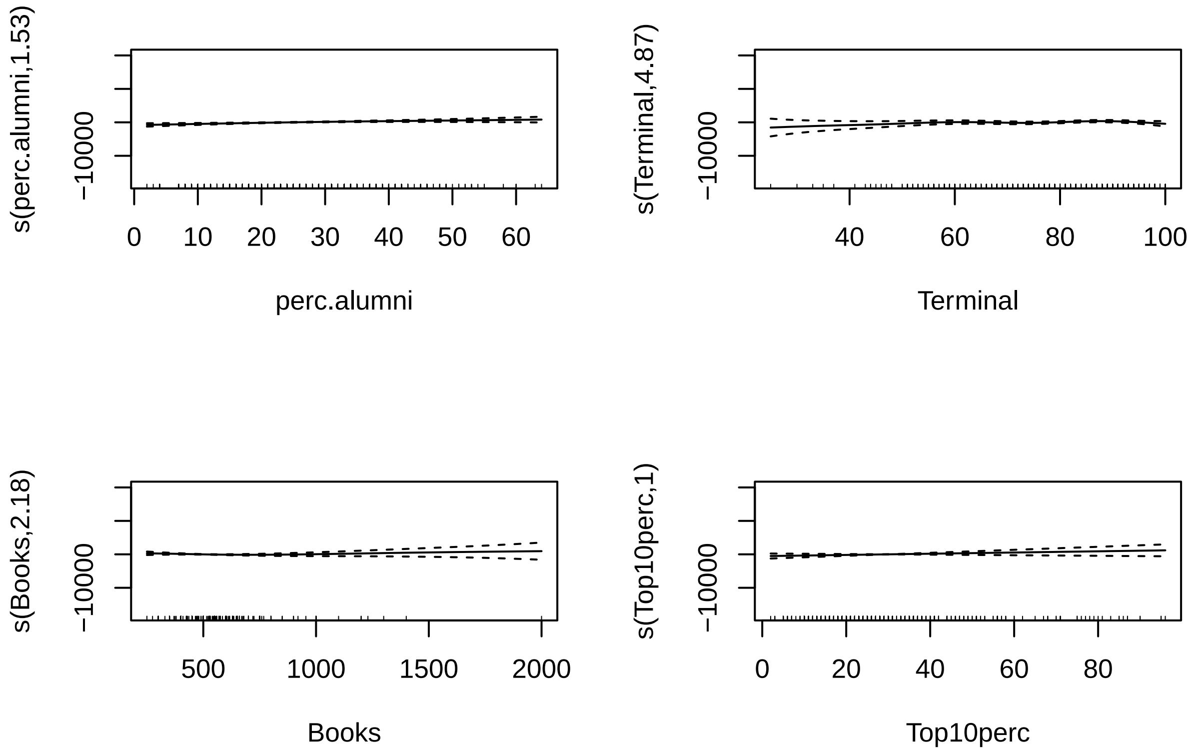
```r
gam.fit$bestTune
```

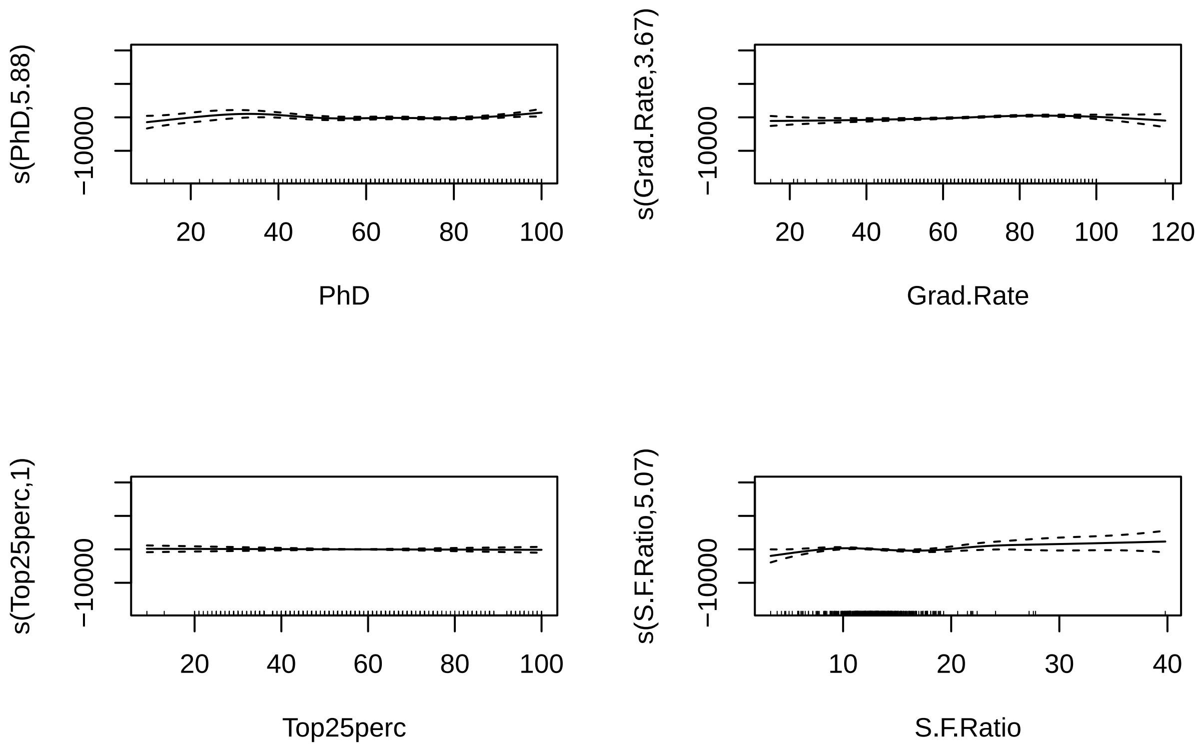
```
##   select method
## 1  FALSE GCV.Cp
```

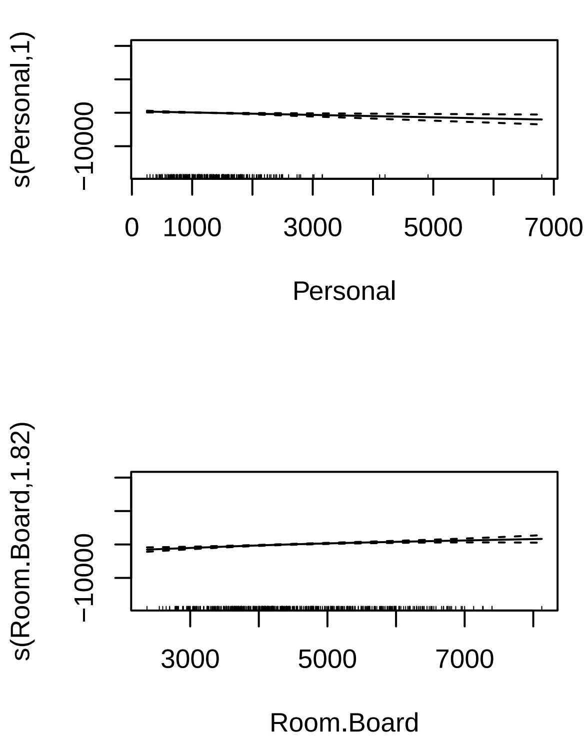
```r
gam.fit$finalModel
```

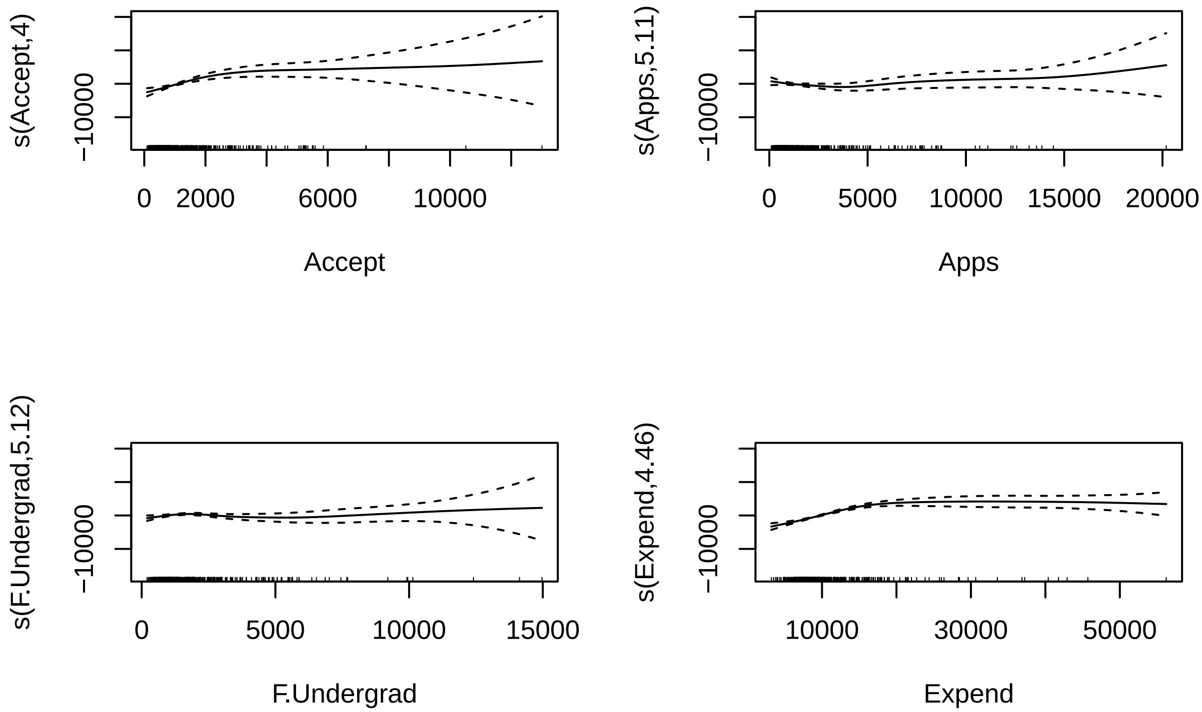
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc.alumni) + s(Terminal) + s(Books) + s(Top10perc) +
##     s(PhD) + s(Grad.Rate) + s(Top25perc) + s(S.F.Ratio) + s(Personal) +
##     s(P.Undergrad) + s(Room.Board) + s(Enroll) + s(Accept) +
##     s(Apps) + s(F.Undergrad) + s(Expend)
##
## Estimated degrees of freedom:
## 1.53 4.87 2.18 1.00 5.88 3.67 1.00
## 5.07 1.00 1.00 1.82 1.00 4.00 5.11
## 5.12 4.46  total = 49.71
##
## GCV score: 2790970
```

```r
par(mfrow = c(2,2))
plot(gam.fit$finalModel)
```


s(perc.alumni,1.53)
−10000
0
10
20
30
40
50
60
perc.alumni
s(Terminal,4.87)
−10000
40
60
80
100
Terminal
s(Books,2.18)
−10000
500
1000
1500
2000
Books
s(Top10perc,1)
−10000
0
20
40
60
80
Top10perc


s(PhD,5.88)
−10000
20
40
60
80
100
PhD
s(Grad.Rate,3.67)
−10000
20
40
60
80
100
120
Grad.Rate
s(Top25perc,1)
−10000
20
40
60
80
100
Top25perc
s(S.F.Ratio,5.07)
−10000
10
20
30
40
S.F.Ratio


s(Personal,1)
−10000
0
1000
3000
5000
7000
Personal
s(P.Undergrad,1)
−10000
0
1000
3000
5000
P.Undergrad
s(Room.Board,1.82)
−10000
3000
5000
7000
Room.Board
s(Enroll,1)
−10000
0
1000
2000
3000
Enroll




The GAM model includes all the predictors. A straight, horizontal line indicates no significant relationship, such as perc.alumni, Terminal, Books, Grad.Rate, Top10perc, PhD, Top25perc, Personal, P.Undergrad, and Room.Board.

However, curves or non-horizontal lines suggest nonlinear associations, like S.F.Ratio, F.Undergrad, Accept, Apps, and Expend.

Also, there are straight non-horizontal lines suggest linearity relationship, such as Enroll.

The test error of gam is

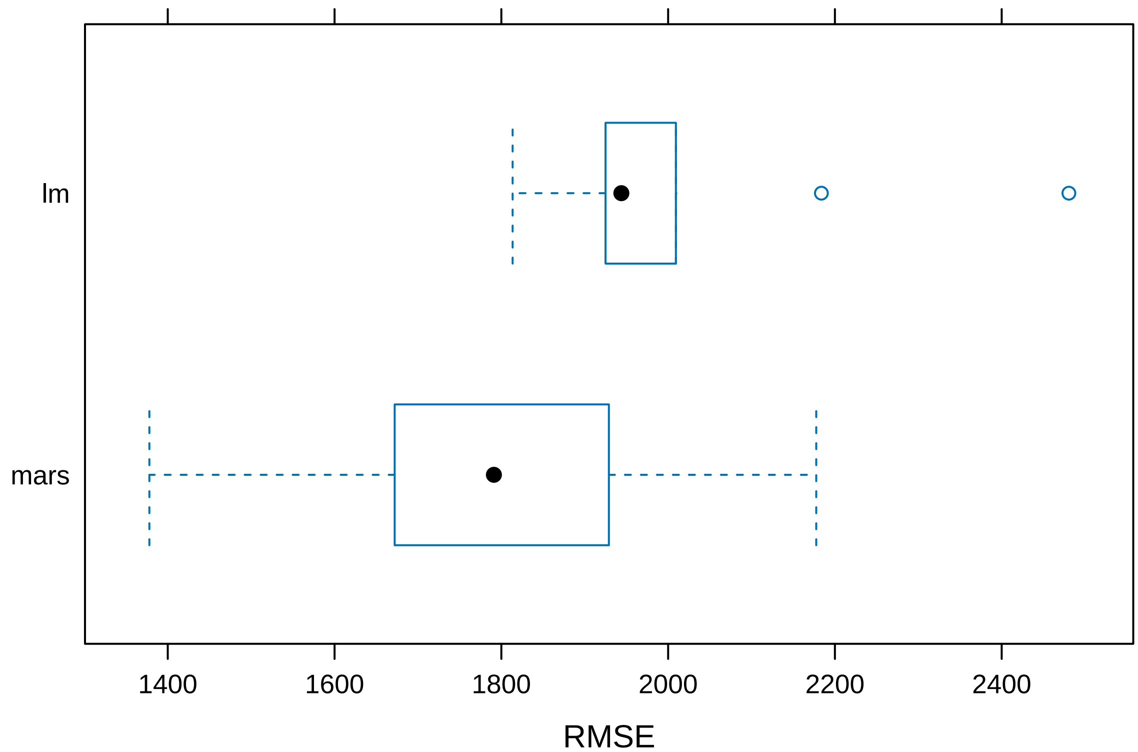
```
gam.pred <- predict(gam.fit, newdata = x_test)
sqrt(mean((y_test - gam.pred)^2))
```

```
## [1] 1722.484
```

(d) In this dataset, would you favor a MARS model over a linear model for predicting out-ofstate tuition? If so,why? More broadly, in general applications, do you consider a MARS model to be superior to a linear model? Please share your reasoning.

```
set.seed(2)
lm.fit = train(x_train, y_train,
                method = "lm",
                trControl = ctrl1)
```

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




Based on this boxplot, the MARS model appears to perform better in terms of having a lower median RMSE, which suggests it is making more accurate predictions on average. A linear model is typically more appropriate when the relationships between the predictors and the response variable are linear. However, when the relationships are not linear or are more complex, MARS model is better.