P8106 HW2

Shihui Zhu

Contents

College Dataset

(a) EDA

Load data set from "College.csv"

```
college <- read_csv("College.csv")[-1] #remove college names
```

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

```
set.seed(1)
rowTrain <- createDataPartition(y = college$Outstate, p = 0.8, list = FALSE)
```

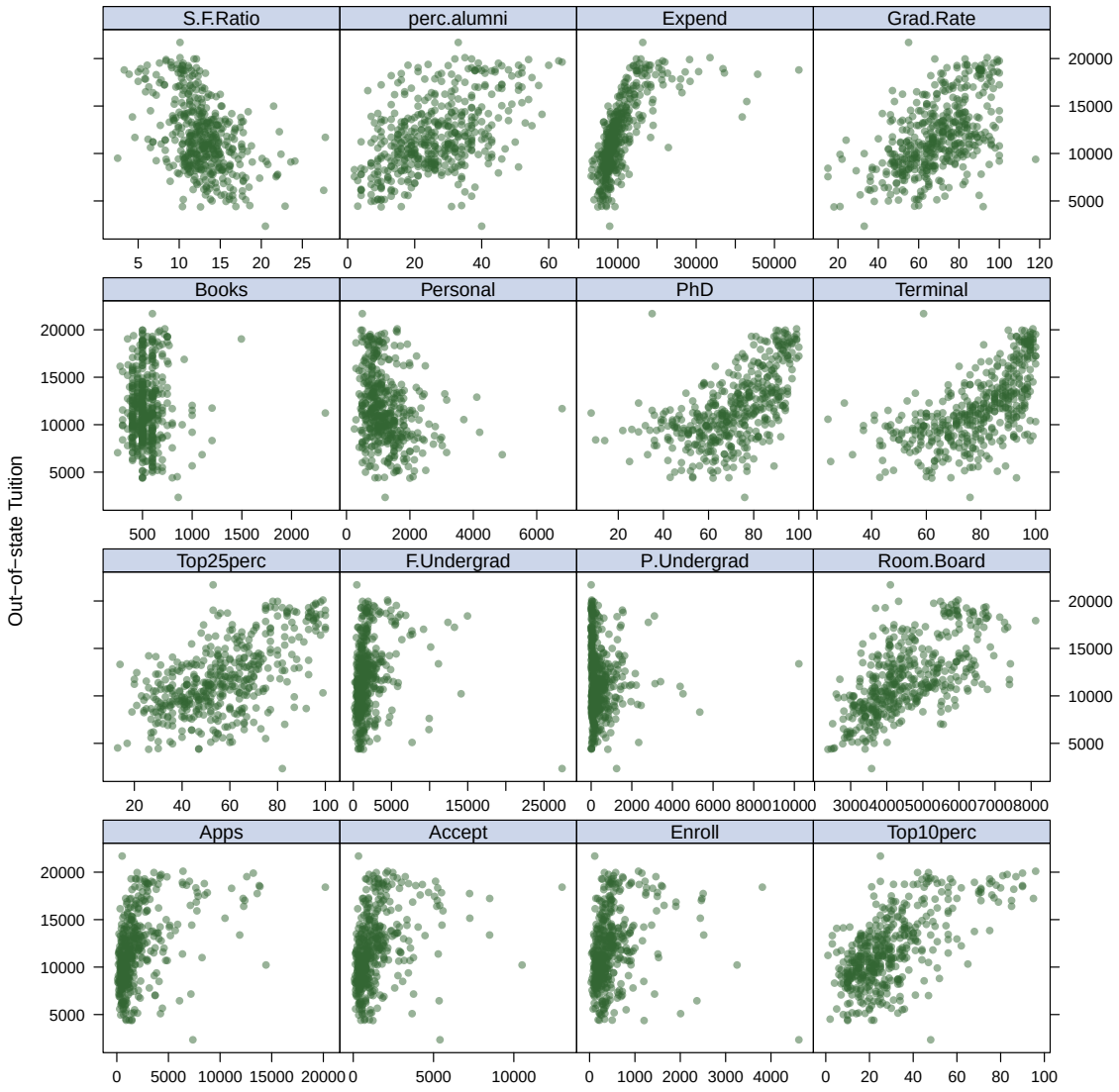
Perform exploratory data analysis using the training data:

```
train.set <- college[rowTrain,]

x <- train.set %>%
  select(-Outstate)
y <- train.set$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)

# Scatter plots
featurePlot(x, y, plot = "scatter", labels = c("","Out-of-state Tuition"),
            type = c("p"), layout = c(4, 4))
```



From the scatter plots above we see that most of the predictors are not linearly associated with response variable (Outstate). For example, data points from plots of *Accept*, *Enroll*, *F.Undergrad*, *P.Undergrad*, *Personal* are clustered in the left side of the plot. This suggests that we may need to use nonlinear model to model our data.

(b) Smoothing Spline Models

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.

For a range of degrees of freedom

df ranges from $(1, nx]$, nx the number of unique x values, in this case, number of unique Terminal values

```
Terminal.grid <- seq(from = min(unique(train.set$Terminal))-10, max(unique(train.set$Terminal))+10, by =

fit.ss <- smooth.spline(train.set$Terminal, train.set$Outstate, lambda = 0.03, cv = FALSE, df = seq(fro
fit.ss$df
```

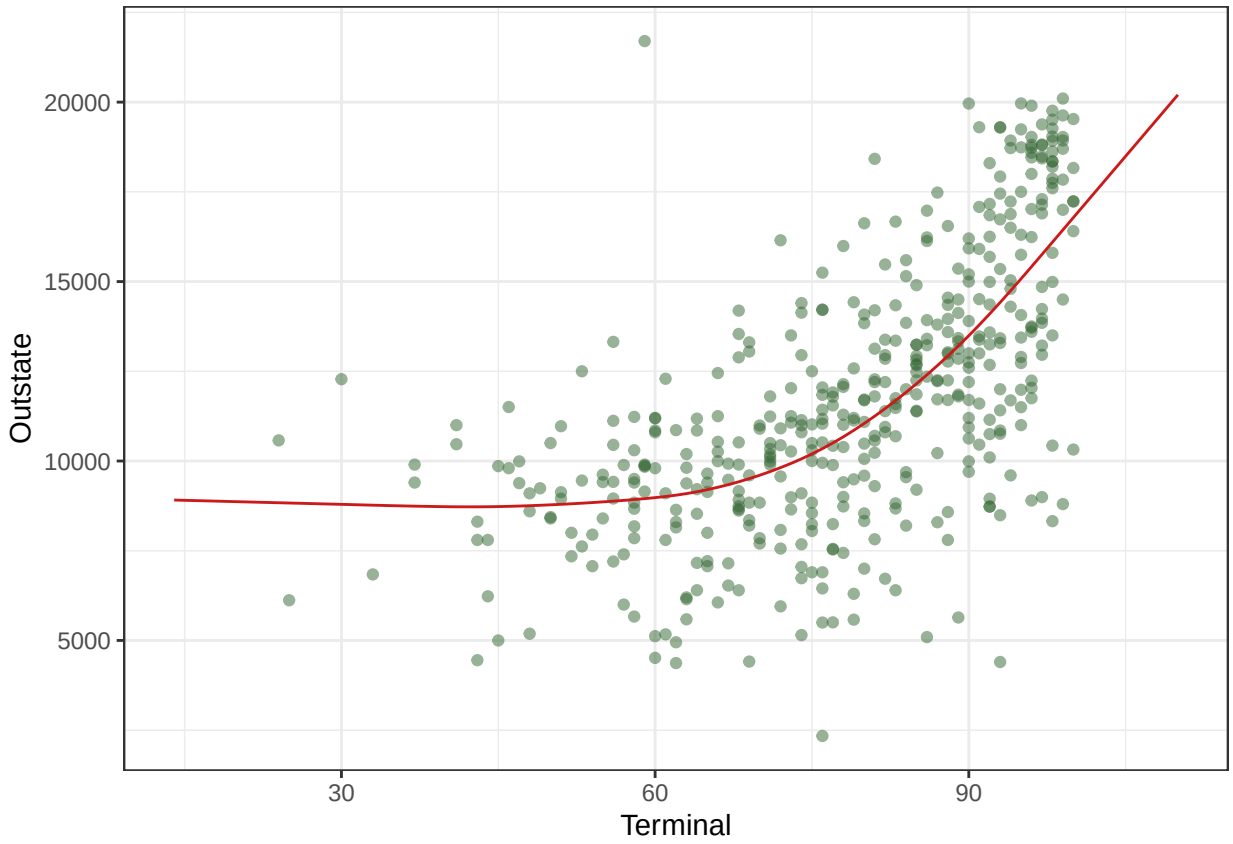
```
## [1] 4.550054
```

```
pred.ss <- predict(fit.ss,
                   x = Terminal.grid)

pred.ss.df <- data.frame(pred = pred.ss$y,
                         terminnal = Terminal.grid)

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

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



The smoothing spline model fitted using a range of degrees of freedom is 4.10501 with $\lambda = 0.03$.

Now we can use cross-validation to select the degrees of freedom:

```
# Use CV
fit.ss.cv <- smooth.spline(train.set$Terminal, train.set$Outstate, cv = TRUE)
fit.ss.cv$df
```

```
## [1] 4.892078
```

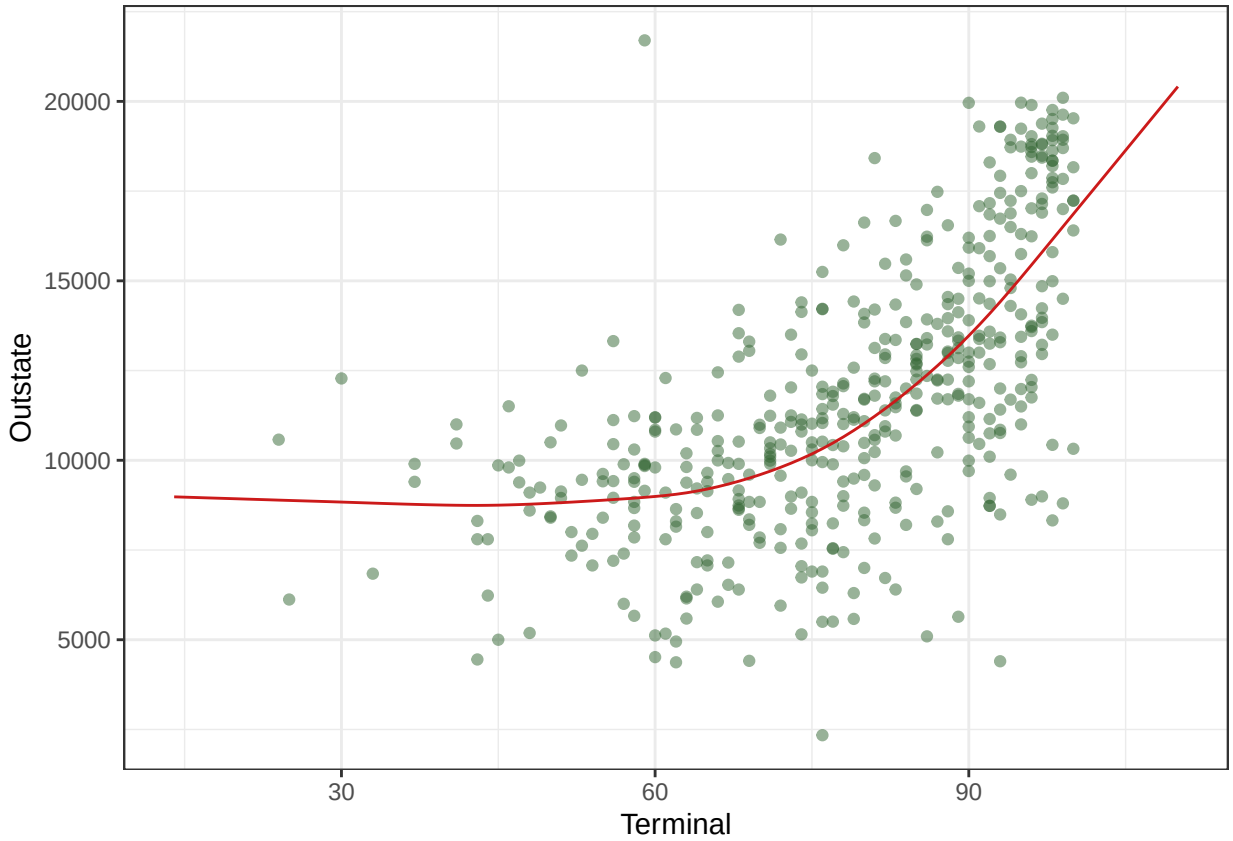
```
fit.ss.cv$lambda
```

```
## [1] 0.0210592
```

```
pred.ss.cv <- predict(fit.ss.cv,
                   x = Terminal.grid)
```

```
pred.ss.df.cv <- data.frame(pred = pred.ss.cv$y,
                            terminnal = Terminal.grid)

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



The smoothing spline model fitted using CV has degrees of freedom is 4.892078 with $\lambda = 0.0210592$.

(c) GAM

Fit GAM using all predictors

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

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Outstate ~ s(Apps) + s(Accept) + s(Enroll) + s(Top10perc) + s(Top25perc) +
##     s(F.Undergrad) + s(P.Undergrad) + s(Room.Board) + s(Books) +
##     s(Personal) + s(PhD) + s(Terminal) + s(S.F.Ratio) + s(perc.alumni) +
##     s(Expend) + s(Grad.Rate)
```

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11779.07      74.68   157.7   <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(Apps)         4.447  5.422  2.510 0.025598 *
## s(Accept)       4.186  5.134  4.088 0.001209 **
## s(Enroll)       1.000  1.000 21.136 6.27e-06 ***
## s(Top10perc)    1.000  1.000  5.263 0.022291 *
## s(Top25perc)    1.000  1.000  1.030 0.310786
## s(F.Undergrad) 5.507  6.536  2.078 0.063787 .
## s(P.Undergrad) 1.000  1.000  1.225 0.269120
## s(Room.Board)  2.472  3.143 14.600  < 2e-16 ***
## s(Books)        2.169  2.706  1.568 0.282200
## s(Personal)     1.000  1.000  4.639 0.031845 *
## s(PhD)          1.806  2.287  0.891 0.446154
## s(Terminal)     1.000  1.000  1.164 0.281302
## s(S.F.Ratio)    3.686  4.647  2.242 0.047853 *
## s(perc.alumni) 6.052  7.162  4.127 0.000229 ***
## s(Expend)       6.868  7.935 19.494  < 2e-16 ***
## s(Grad.Rate)    3.556  4.470  2.816 0.022655 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.819   Deviance explained = 83.7%
## GCV = 2.8242e+06  Scale est. = 2.5265e+06  n = 453
```

```
gam.full$df.residual
```

```
## [1] 405.2527
```

```
# Training RMSE
sqrt(mean(residuals.gam(gam.full,type="response")^2))
```

```
## [1] 1503.405
```

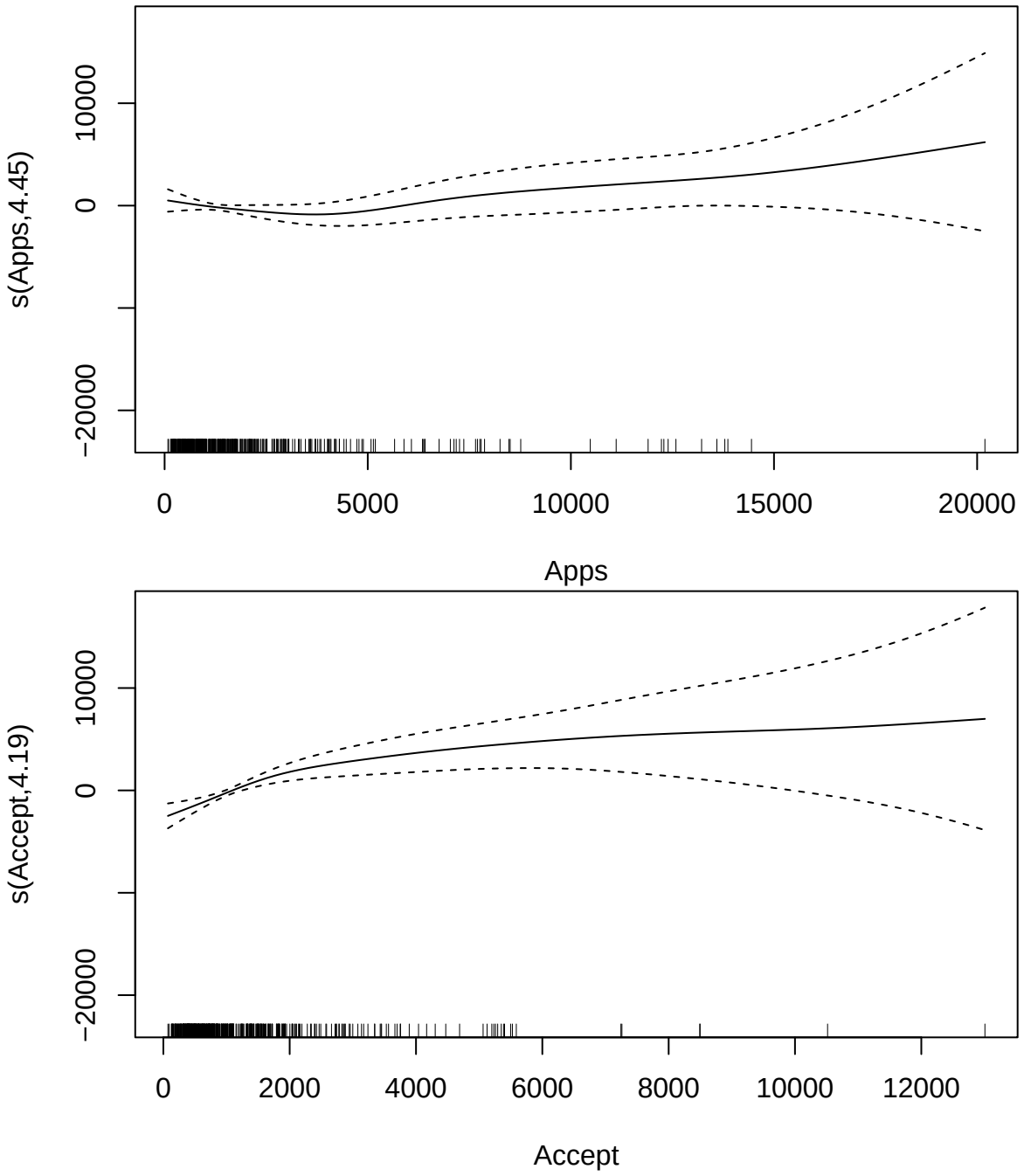
The total degrees of freedom of the GAM model is 405.2527. The p-value of some of the predictors show that the predictor might not be significant: Top25perc, F.Undergrad, P.Undergrad, Books, PhD, and Terminal. Also, among the significant predictors, some of the them are likely to have linear relationship with the model: Enroll, Top10perc, and Personal.

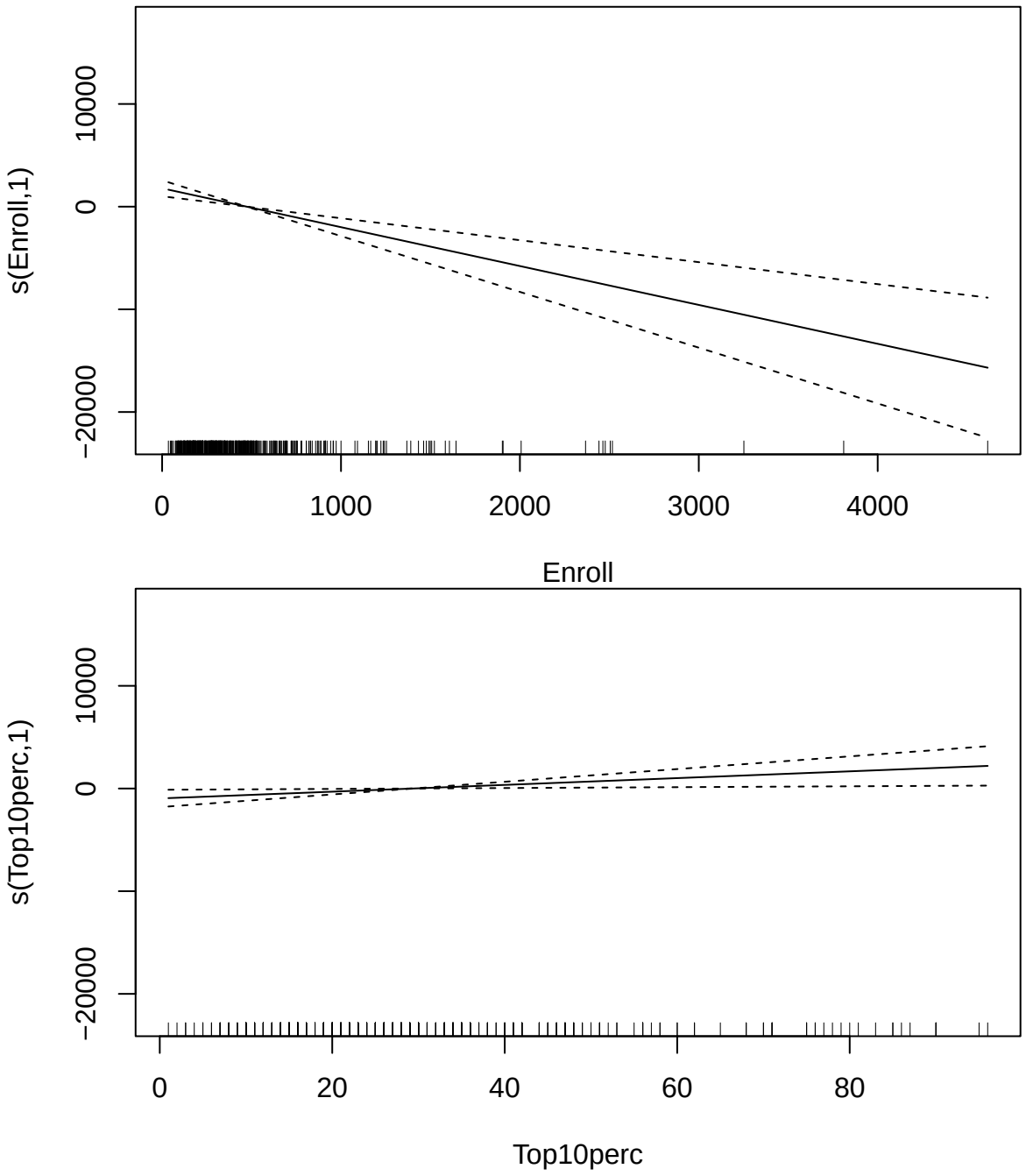
The deviance explained by the model is 83.7%, and the adjusted R-squared is 0.819, which means the model explains the data well. The RMSE os the model is 1503.405.

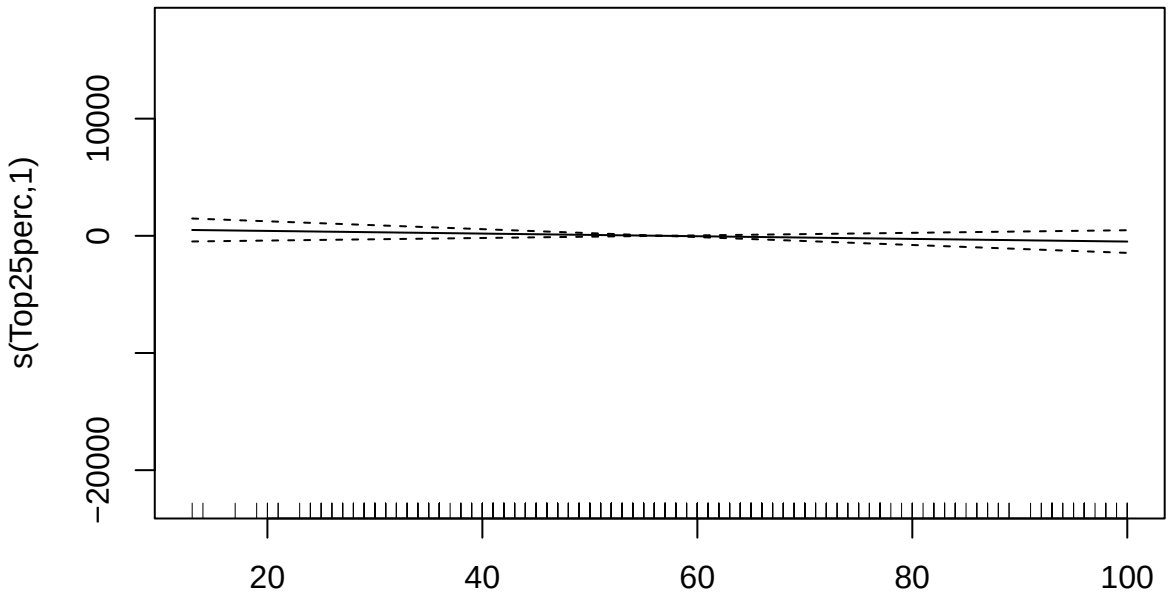
Plot results:

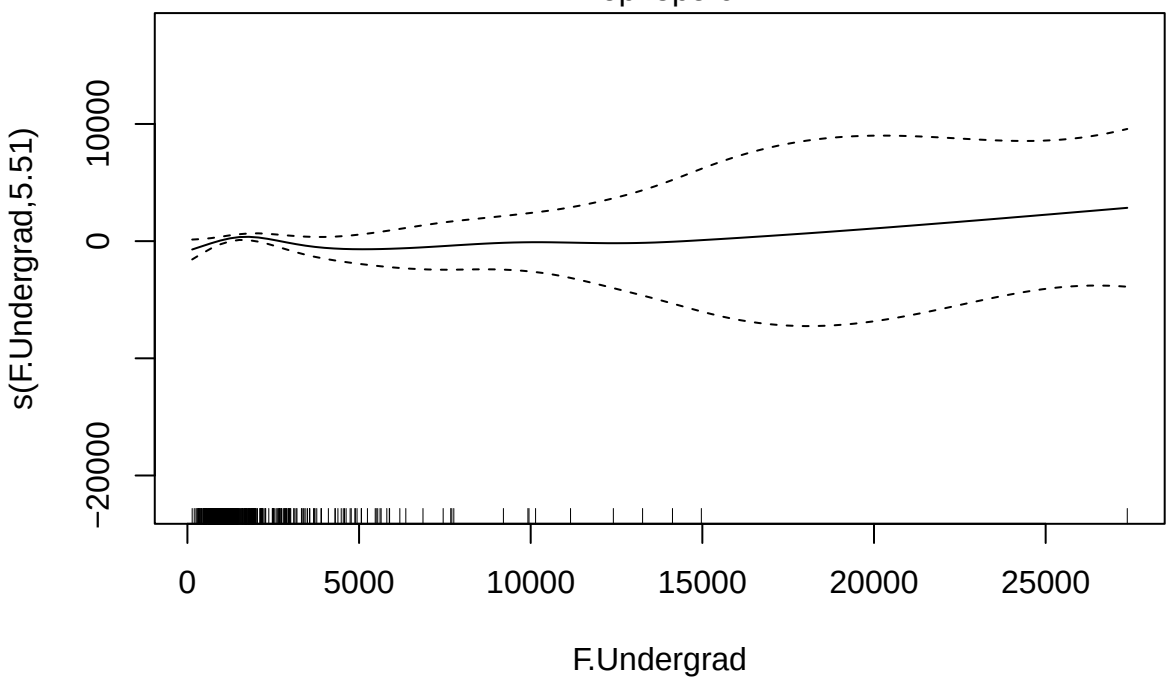
The plots of each predictor v.s. the response (Outstate) shown below:

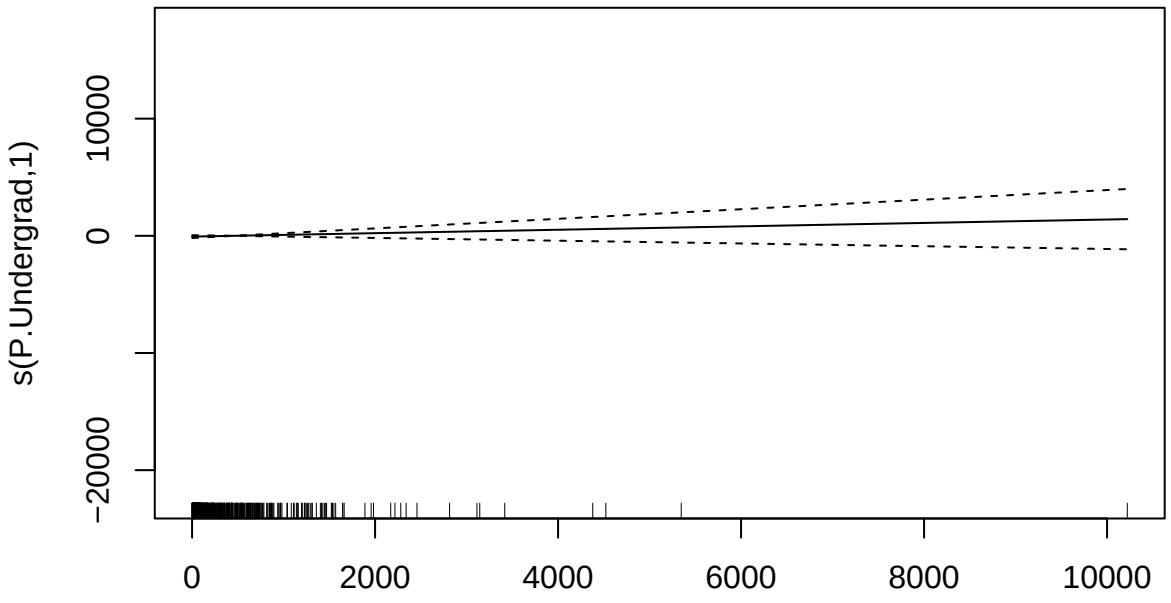
```
plot(gam.full)
```

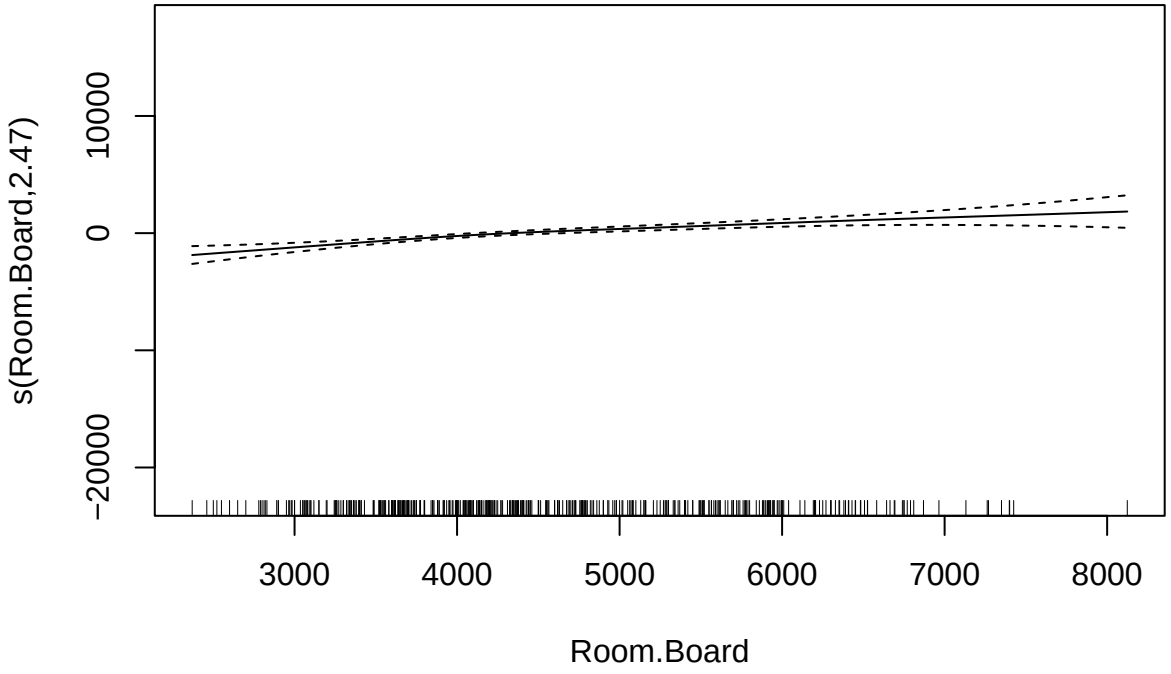


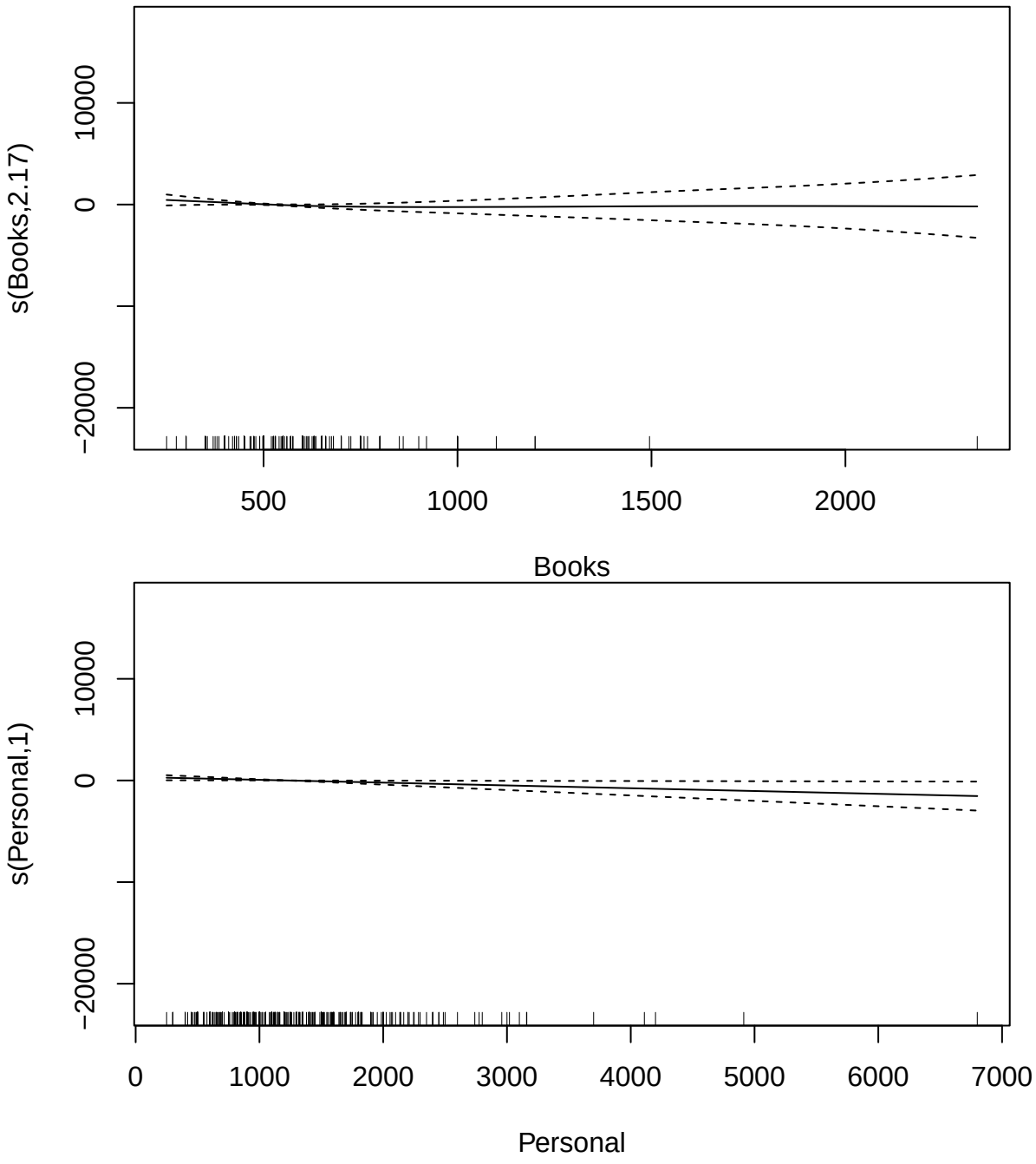

s(Enroll,1)
10000
0
−20000
0
1000
2000
3000
4000
Enroll
s(Top10perc,1)
10000
0
−20000
0
20
40
60
80
Top10perc

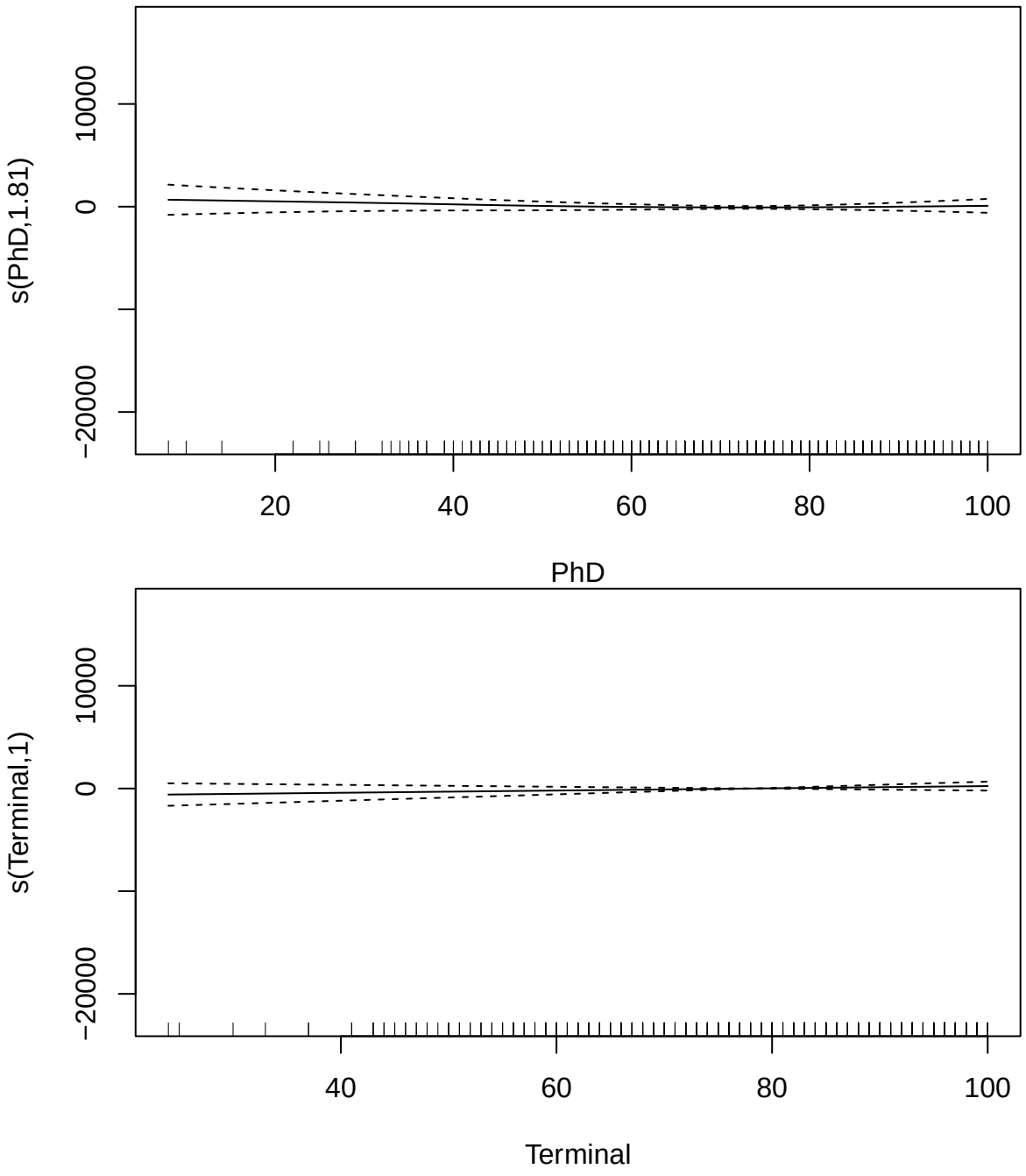

s(Top25perc,1)
10000
0
−20000
20
40
60
80
100
Top25perc

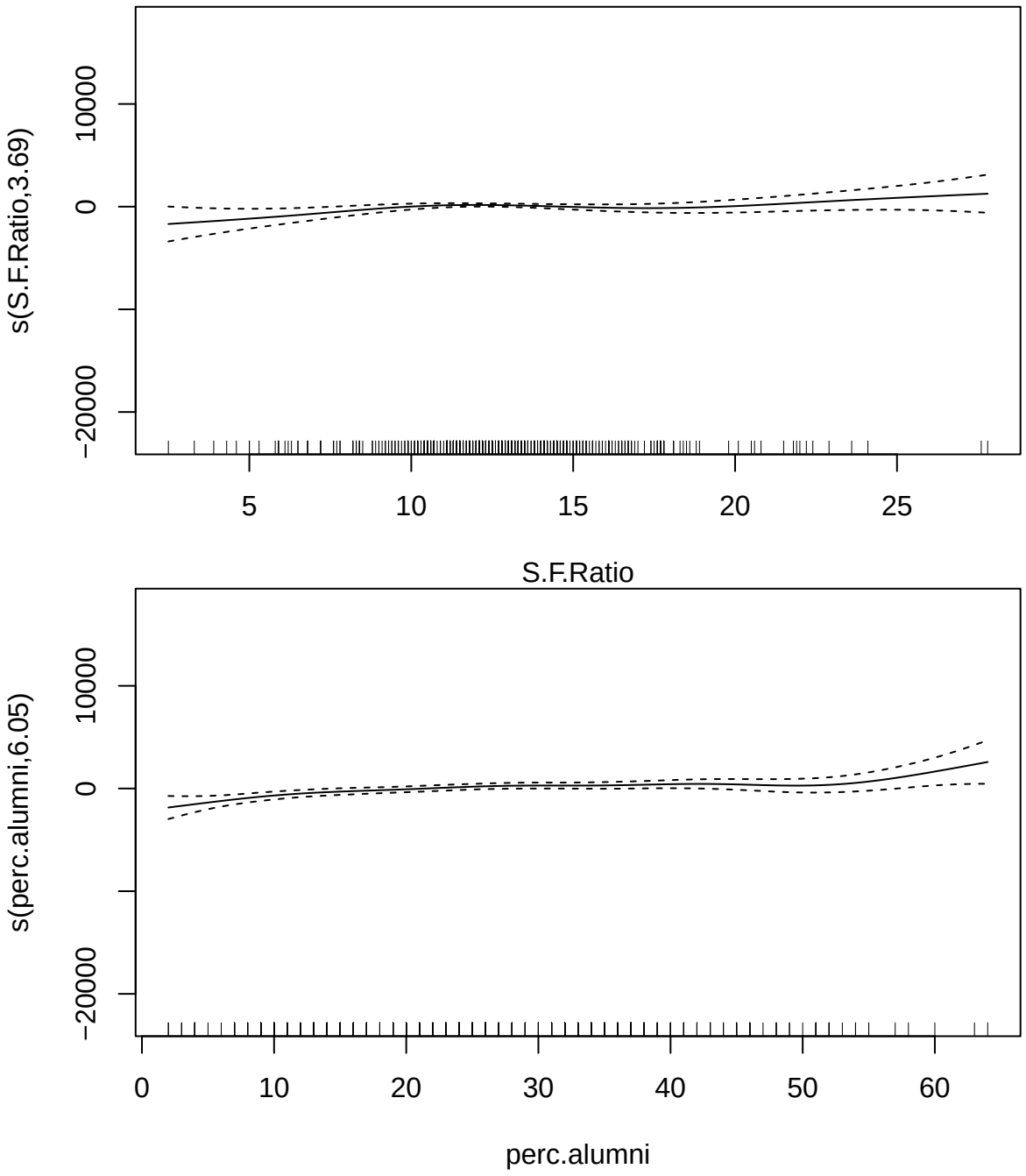

s(F.Undergrad,5.51)
10000
0
−20000
0
5000
10000
15000
20000
25000
F.Undergrad

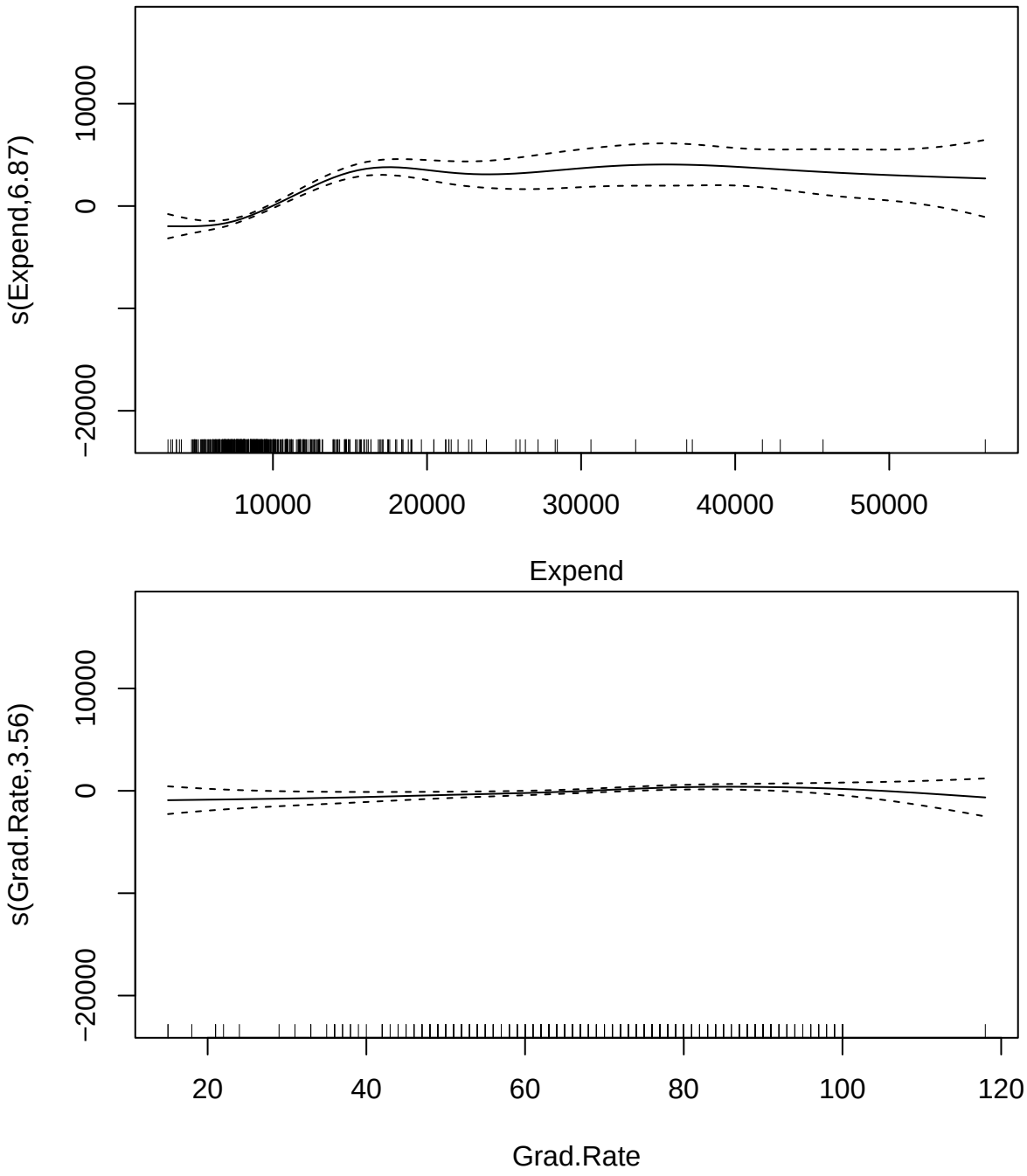

s(P.Undergrad,1)
10000
0
−20000
0
2000
4000
6000
8000
10000
P.Undergrad


s(Room.Board,2.47)
10000
0
−20000
3000
4000
5000
6000
7000
8000
Room.Board


s(Books,2.17)
10000
0
−20000
500
1000
1500
2000
Books
s(Personal,1)
10000
0
−20000
0
1000
2000
3000
4000
5000
6000
7000
Personal


s(PhD,1.81)
10000
0
−20000
20
40
60
80
100
PhD
s(Terminal,1)
10000
0
−20000
40
60
80
100
Terminal


s(S.F.Ratio,3.69)
10000
0
−20000
5
10
15
20
25
S.F.Ratio
s(perc.alumni,6.05)
10000
0
−20000
0
10
20
30
40
50
60
perc.alumni



Test Error

```
gam.pred  <- predict(gam.full, newdata = college[-rowTrain,])
## Test Error (MSE)
t.mse <- mean((college[-rowTrain,]$Outstate - gam.pred)^2);t.mse
```

```
## [1] 3012372
```

The test error (MSE) of the GAM model is 3012372.

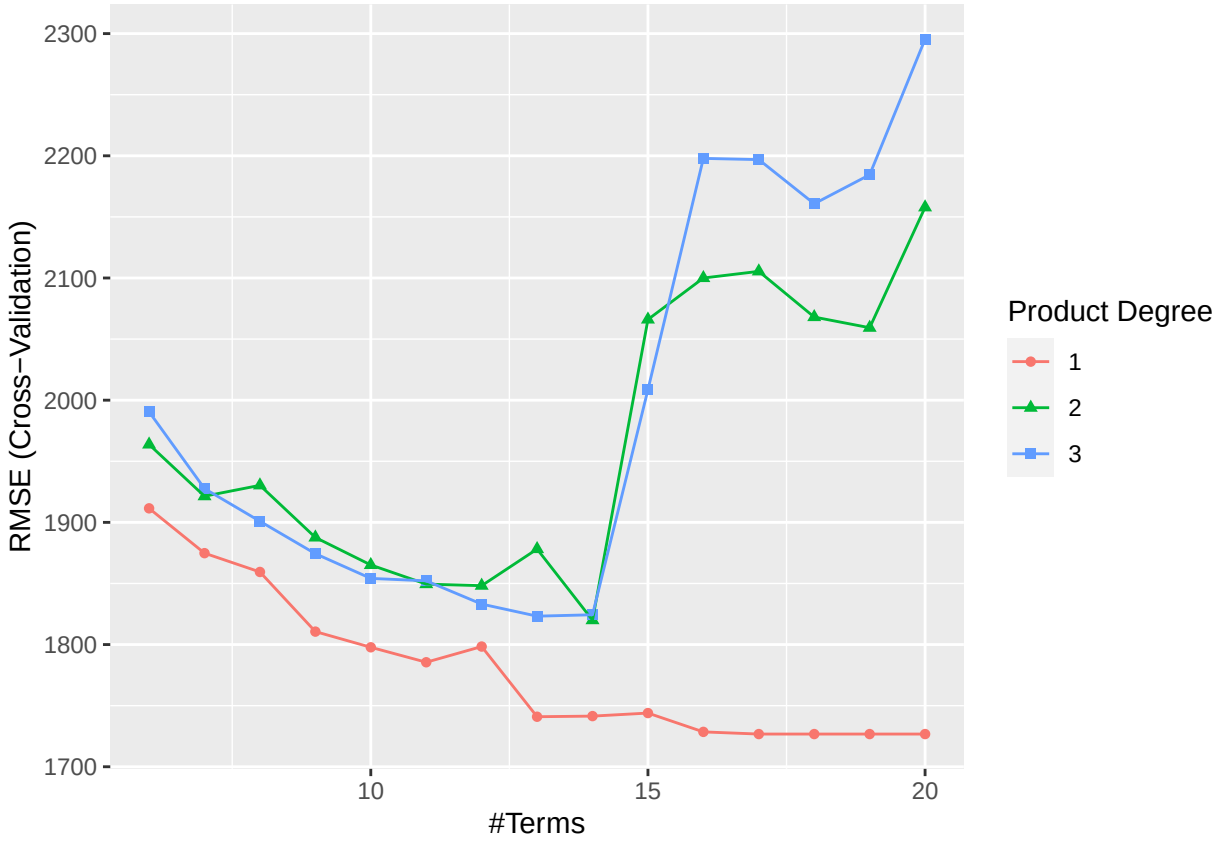
(d) MARS

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.

Build the MARS model

```
ctrl1 <- trainControl(method = "cv", number = 10)
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 6:20)

set.seed(2)
mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)
## Plot of grid tunning
ggplot(mars.fit)
```



The final model is:

```
mars.fit$bestTune
```

```
##    nprune degree
## 12     17      1
```

```
## Coefficient of the MARS model
coef(mars.fit$finalModel)
```

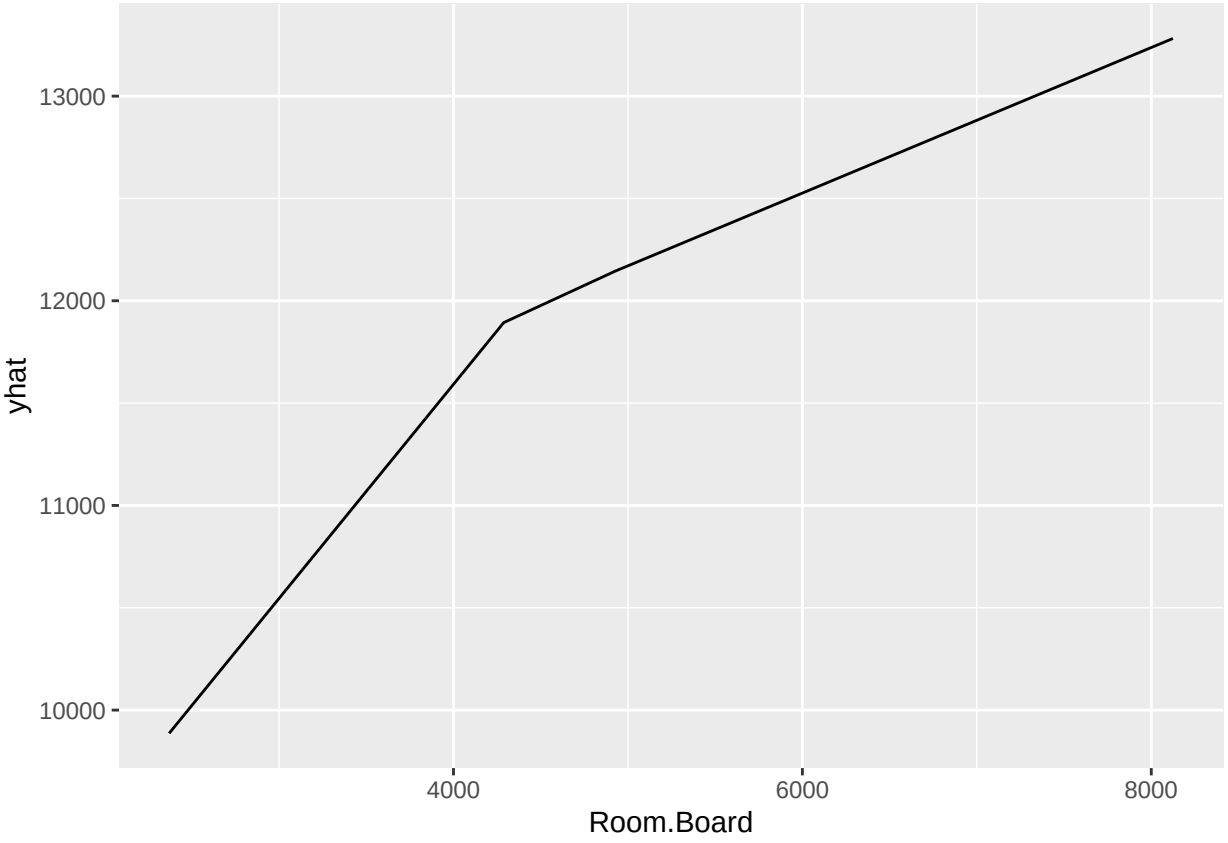
```
##         (Intercept)     h(Expend-15886)     h(79-Grad.Rate)  h(Room.Board-4323)
##        9750.9084463          -0.7366761         -27.4149388           0.3555943
##  h(4323-Room.Board) h(1379-F.Undergrad)   h(22-perc.alumni)        h(Apps-3712)
##          -1.0463218          -1.5733517         -91.7755202           0.4447256
##    h(1300-Personal)      h(Expend-6897)       h(Enroll-911)       h(911-Enroll)
##           0.8665098           0.7149307          -2.0263362           5.7508922
##      h(2109-Accept)
##          -1.9904298
```

The optimal model with minimum prediction error has 17 retained terms, and 1 degree of interaction.

Produce the PDP plots

PDP of Room.Board predictor

```
pdp::partial(mars.fit, pred.var = c("Room.Board"), grid.resolution = 10) %>% autoplot()
```



Test Error

```
mars.pred  <- predict(mars.fit, newdata = college[-rowTrain,])
## Test Error (MSE)
t.mse <- mean((college[-rowTrain,]$Outstate - mars.pred)^2);t.mse
```

```
## [1] 2774623
```

The test error (MSE) of the MARS model is 2774623.

(e) Model Comparision

According to (c) and (d), we found that the test error of GAM model is 3012372, and the test error of MARS model is 2774623. For data prediction, we want to choose the model with the smaller test error, so we choose MARS model for out-of-state prediction.