DSII_HW2_yc4384

Yangyang Chen

Contents

College Dataset

In this exercise, we explore the application of nonlinear models to analyze the "College" dataset, comprising statistics from 565 US colleges as reported in a past issue of US News and World Report. The response variable is the out-of-state tuition (Outstate). The predictors are

EDA

Load data set from "College.csv"

```
college_df <- readr::read_csv("College.csv")[-1] |>
  janitor::clean_names()#remove college names
```

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

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

Perform exploratory data analysis using the training data:

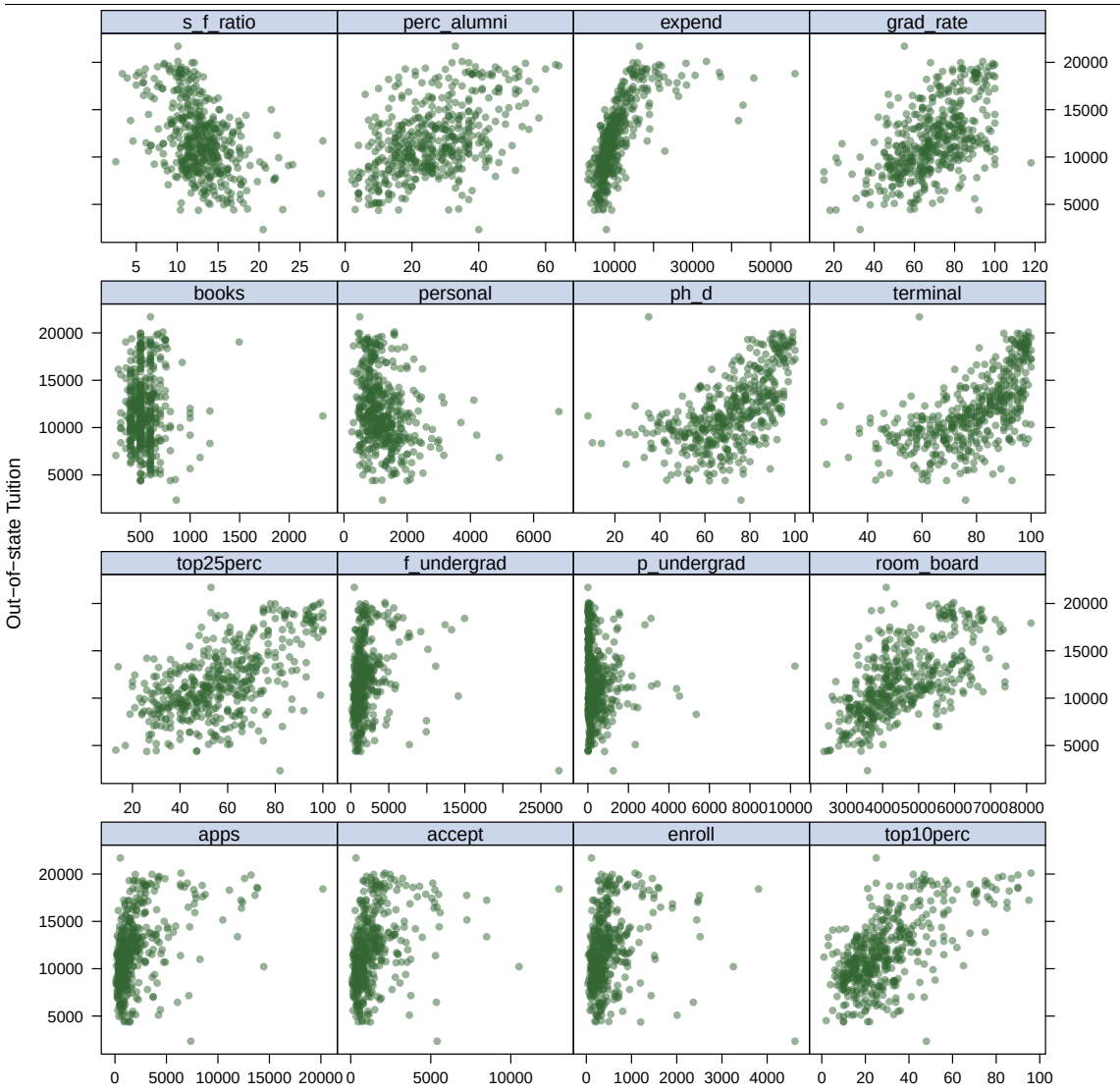
```
train.set <- college_df[rowTrain,]

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))
```

(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. 3



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.

(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.

Range of degrees of freedom

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

```
perc_alumni.grid <- seq(from = min(unique(train.set$perc_alumni))-10, to=max(unique(train.set$perc_alumn

fit.ss <- smooth.spline(train.set$perc_alumni, train.set$outstate, lambda = 0.03, cv = FALSE)
fit.ss$df
```

```
## [1] 4.59127
```

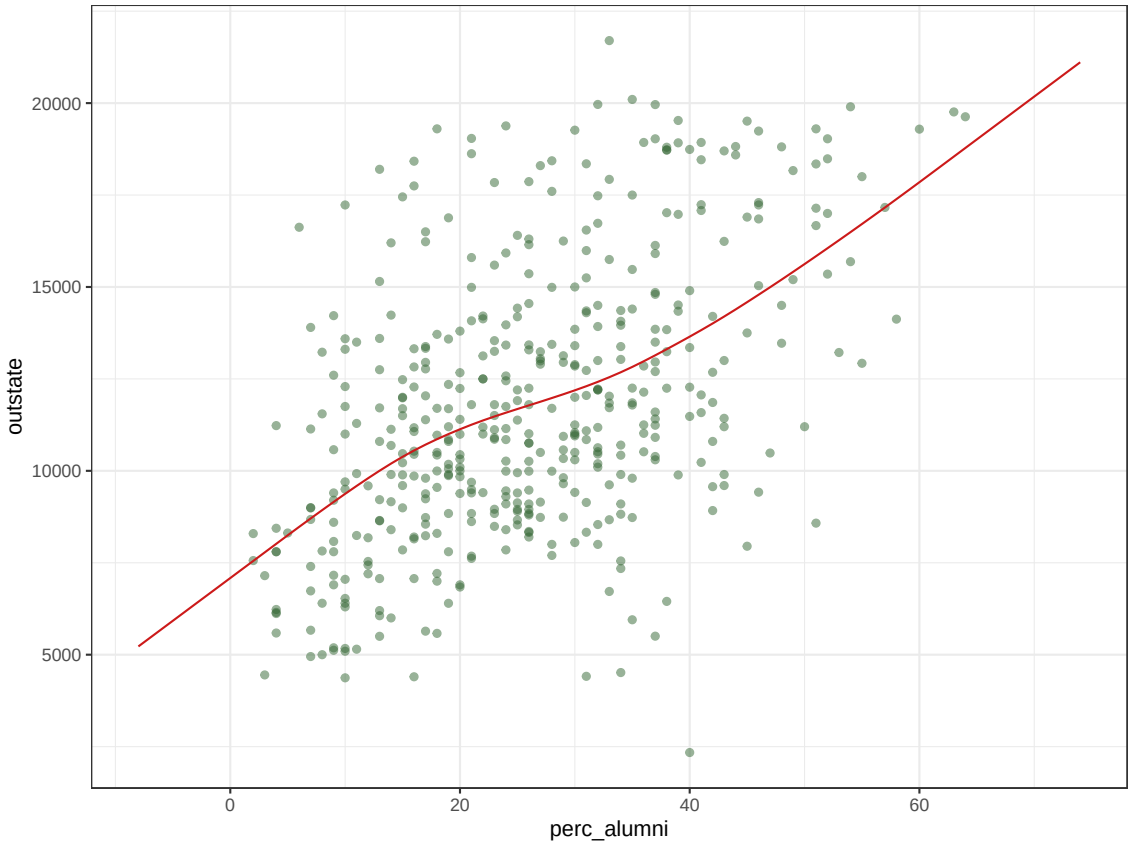
(a) Fit smoothing spline models to predict out-of-state tuition (Outstate) using the percentage of alumni who donate (perc.alumni) as the only predictor, across a range of degrees of freedom. Plot the model fits for each degree of freedom. 4

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

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

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

p +
geom_line(aes(x = perc_alumni.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 4.59127 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$perc_alumni, train.set$outstate, cv = TRUE)
fit.ss.cv$df
```

```
## [1] 4.508428
```

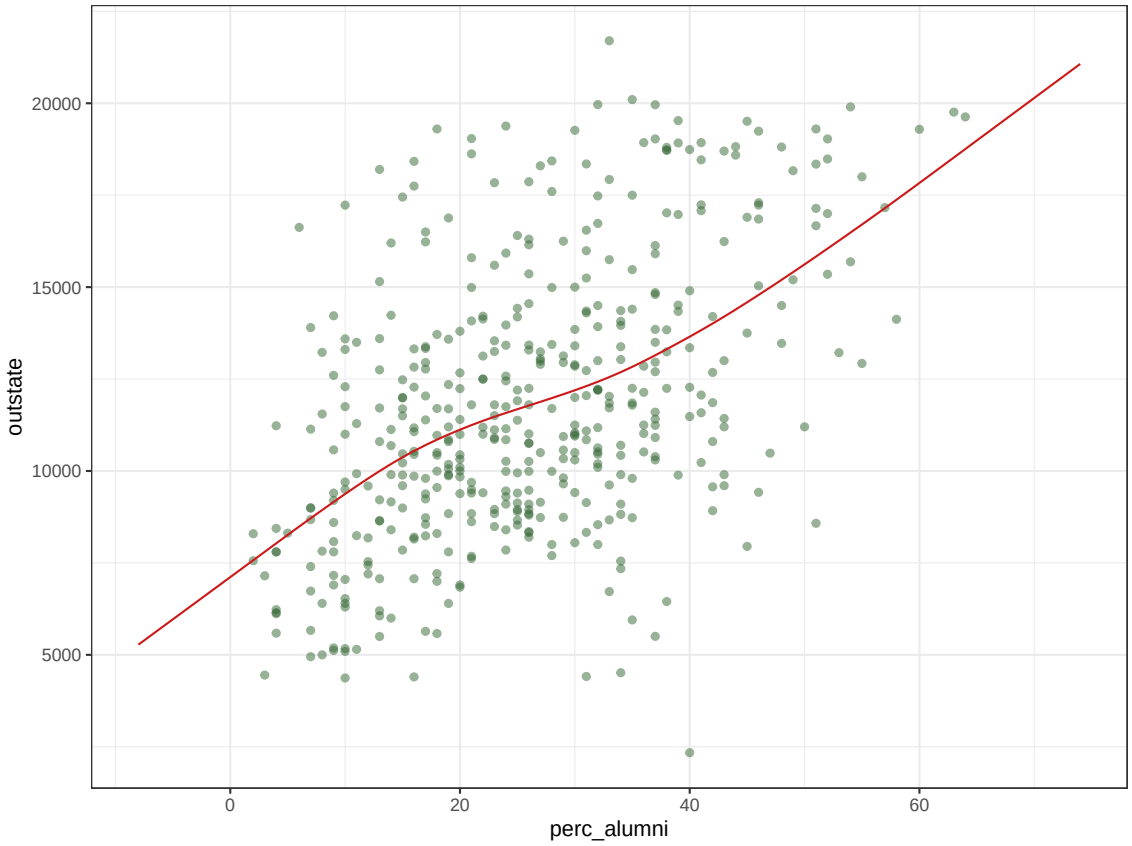
```
fit.ss.cv$lambda
```

```
## [1] 0.03274646
```

```
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)

p +
geom_line(aes(x = perc_alumni.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 4.508428 with $\lambda = 0.03274646$.

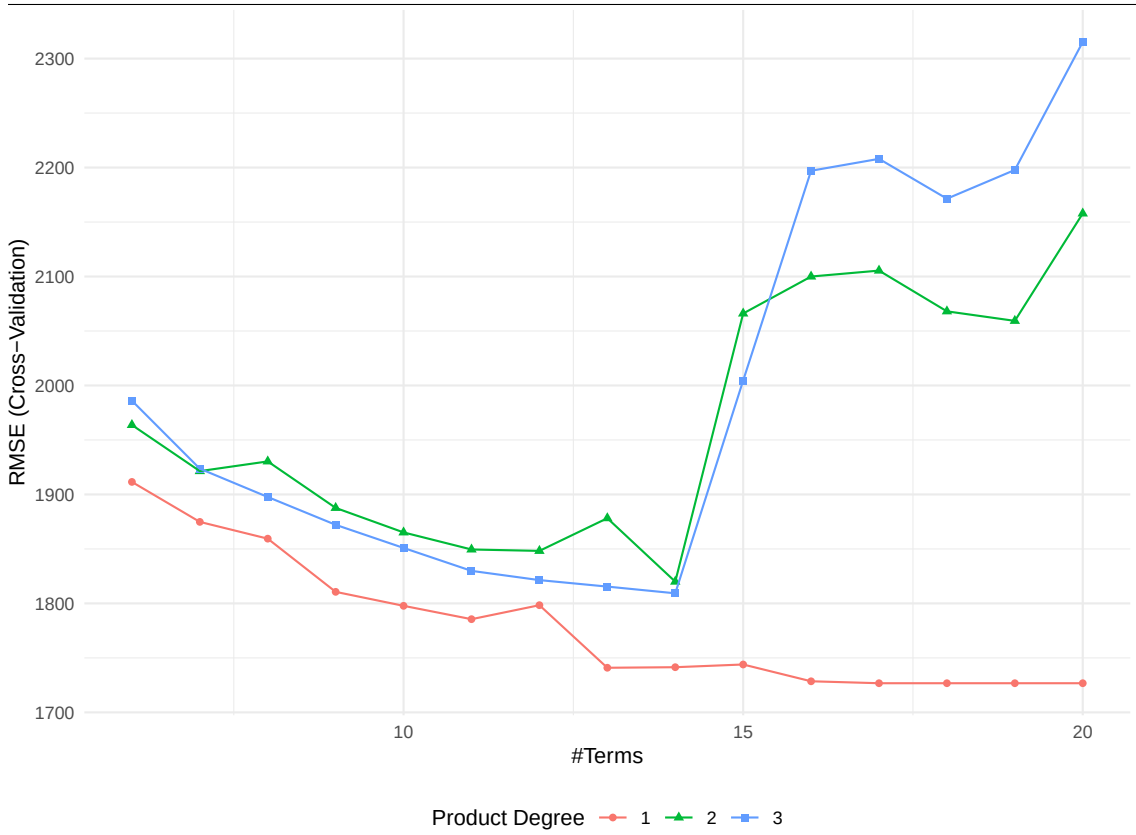
(b) Train a multivariate adaptive regression spline (MARS) model using all the predictors. Report the final model. Present the partial dependence plot of an arbitrary predictor in your final model. Report the test error.

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)
```

(b) 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. 6



The final model is:

```
mars.fit$bestTune
```

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

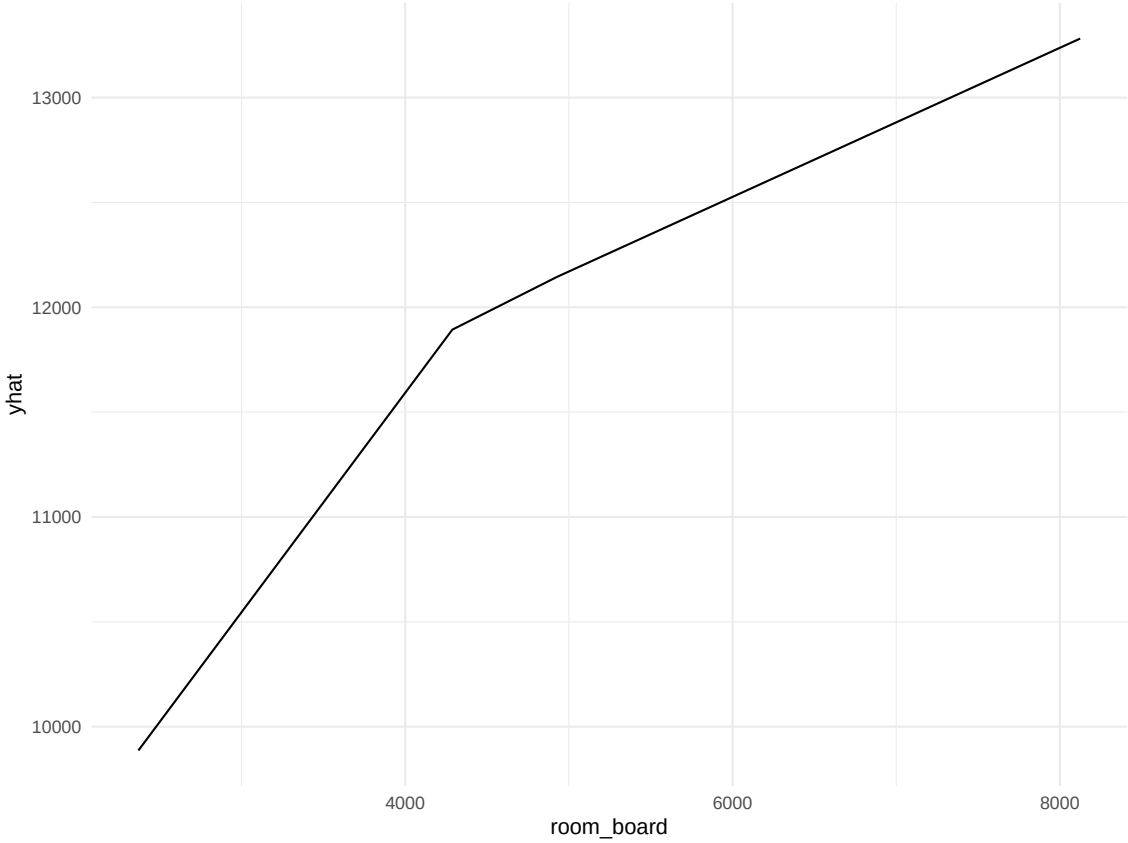
```
##          (Intercept)     h(expend-15886)     h(79-grad_rate)  h(room_board-4323)
##        9750.9084463          -0.7366761         -27.4149388           0.3555943
##  h(4323-room_board) h(1379-f_undergrad)   h(22-perc_alumni)        h(apps-3712)
##          -1.0463218          -1.5733517         -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_df[-rowTrain,])
## Test Error (MSE)
t.mse =
  mean((college_df[-rowTrain,]$outstate - mars.pred)^2)
t.mse
```

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

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

(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.

Fit GAM using all predictors

```
gam.full =
  train.set |>
  gam(outstate ~ s(apps)+s(accept)+s(enroll)+s(top10perc)+s(top25perc)+s(f_undergrad)+s(p_undergrad)+s(

summary(gam.full)
```

```
##
```

(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. 8

```
## 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(ph_d) + 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(ph_d)         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`, `ph_d`,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:

(d) In this dataset, would you favor a MARS model over a linear model for predicting out-of- state 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. 9

The plots of each predictor v.s. the response (`outstate`) shown inthe pdf named `gam.full.pdf` file:

```
# Open a PDF device with specified width and height
pdf("gam_plot.pdf", width = 6, height = 4)

# Plot the GAM model
plot(gam.full)

# Close the PDF device
dev.off()
```

```
## pdf
##   2
```

Test Error

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

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

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

(d) In this dataset, would you favor a MARS model over a linear model for predicting out-of- state 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.

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.