Assignment 2

Zheyan Liu

Contents

Intro and data preparation

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).

Read data

Drop college column

```
df =
  read_csv('data/College.csv', show_col_types = FALSE) %>%
  janitor::clean_names() %>%
  select(-college)
```

Split the dataset into training and testing

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

```
trainRows <- createDataPartition(y = df$outstate, p = 0.8, list = FALSE)
training_df = df[trainRows, ]
testing_df = df[-trainRows, ]

x_train <- model.matrix(outstate~.,training_df)[,-1]
y_train <- training_df$outstate

x_test <- model.matrix(outstate~.,testing_df)[,-1]
y_test <- testing_df$outstate
```

(a) Perform exploratory data analysis using the training data

There are 17 variables in the data and 453 observations.

summary statistics

All variables are continuous

```
summary(training_df)
```

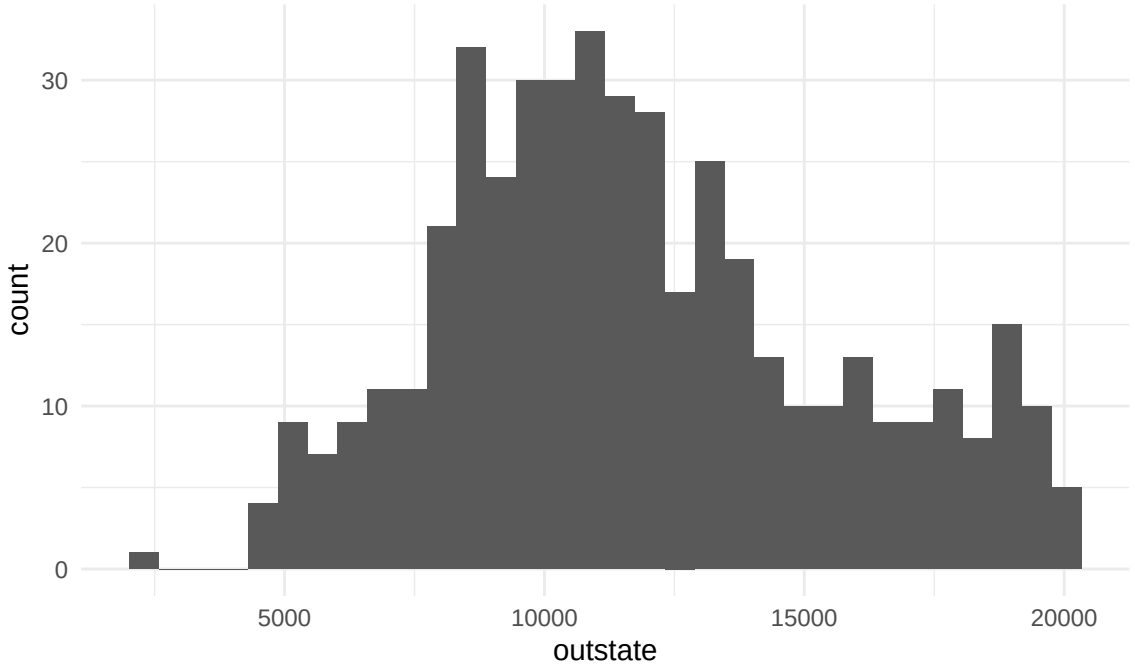
```
##       apps           accept          enroll         top10perc
##  Min.   :   81   Min.   :   72   Min.   :  35.0   Min.   : 1.00
##  1st Qu.:  627   1st Qu.:  503   1st Qu.: 207.0   1st Qu.:16.00
##  Median : 1130   Median :  859   Median : 328.0   Median :25.00
##  Mean   : 2035   Mean   : 1337   Mean   : 465.2   Mean   :29.06
##  3rd Qu.: 2308   3rd Qu.: 1698   3rd Qu.: 523.0   3rd Qu.:36.00
##  Max.   :14446   Max.   :10516   Max.   :4615.0   Max.   :96.00
##    top25perc       f_undergrad     p_undergrad         outstate
##  Min.   :  9.00   Min.   :  139   Min.   :    1.0   Min.   : 2340
##  1st Qu.: 42.00   1st Qu.:  836   1st Qu.:   74.0   1st Qu.: 9100
##  Median : 55.00   Median : 1306   Median :  217.0   Median :11200
##  Mean   : 56.72   Mean   : 1925   Mean   :  455.1   Mean   :11801
```

```
##  3rd Qu.: 69.00   3rd Qu.: 2110   3rd Qu.:  549.0   3rd Qu.:13970
##  Max.   :100.00   Max.   :27378   Max.   :10221.0   Max.   :20100
##    room_board       books           personal         ph_d
##  Min.   :2370   Min.   : 250.0   Min.   : 250   Min.   :  8.00
##  1st Qu.:3730   1st Qu.: 450.0   1st Qu.: 800   1st Qu.: 60.00
##  Median :4400   Median : 500.0   Median :1100   Median : 74.00
##  Mean   :4596   Mean   : 547.7   Mean   :1195   Mean   : 71.24
##  3rd Qu.:5400   3rd Qu.: 600.0   3rd Qu.:1500   3rd Qu.: 86.00
##  Max.   :8124   Max.   :2340.0   Max.   :4913   Max.   :100.00
##     terminal       s_f_ratio      perc_alumni        expend
##  Min.   : 24.0   Min.   : 2.50   Min.   : 2.00   Min.   : 3186
##  1st Qu.: 68.0   1st Qu.:11.20   1st Qu.:16.00   1st Qu.: 7440
##  Median : 81.0   Median :12.80   Median :26.00   Median : 8946
##  Mean   : 78.7   Mean   :13.02   Mean   :25.76   Mean   :10439
##  3rd Qu.: 92.0   3rd Qu.:14.50   3rd Qu.:34.00   3rd Qu.:11361
##  Max.   :100.0   Max.   :39.80   Max.   :63.00   Max.   :56233
##    grad_rate
##  Min.   : 15.00
##  1st Qu.: 58.00
##  Median : 70.00
##  Mean   : 69.03
##  3rd Qu.: 81.00
##  Max.   :118.00
```

histogram of response variable

Distribution of outstate is close to normal distribution, much outstate is around 10000 except a second peak around 17500

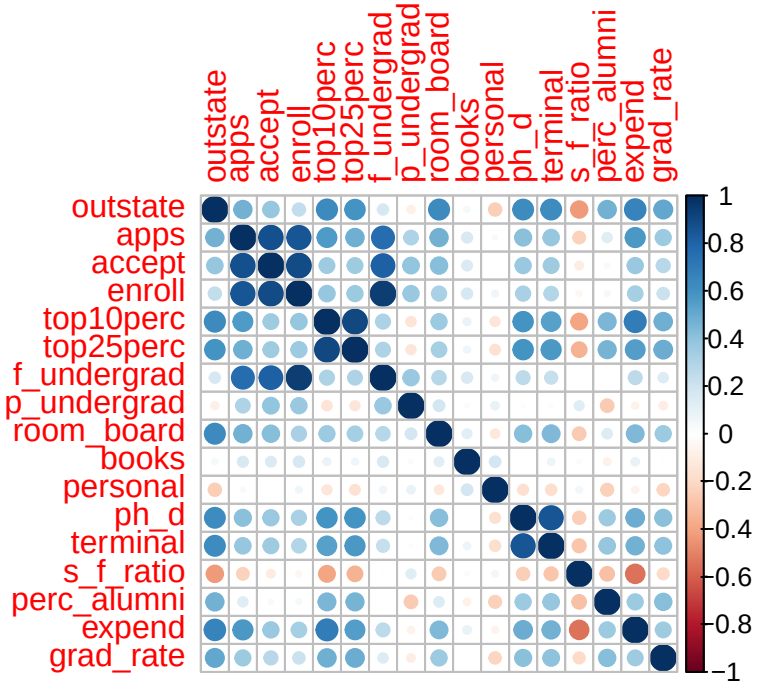
```
ggplot(training_df, aes(x=outstate)) +
  geom_histogram(bins = 32)
```



correlation of response vs. predictors

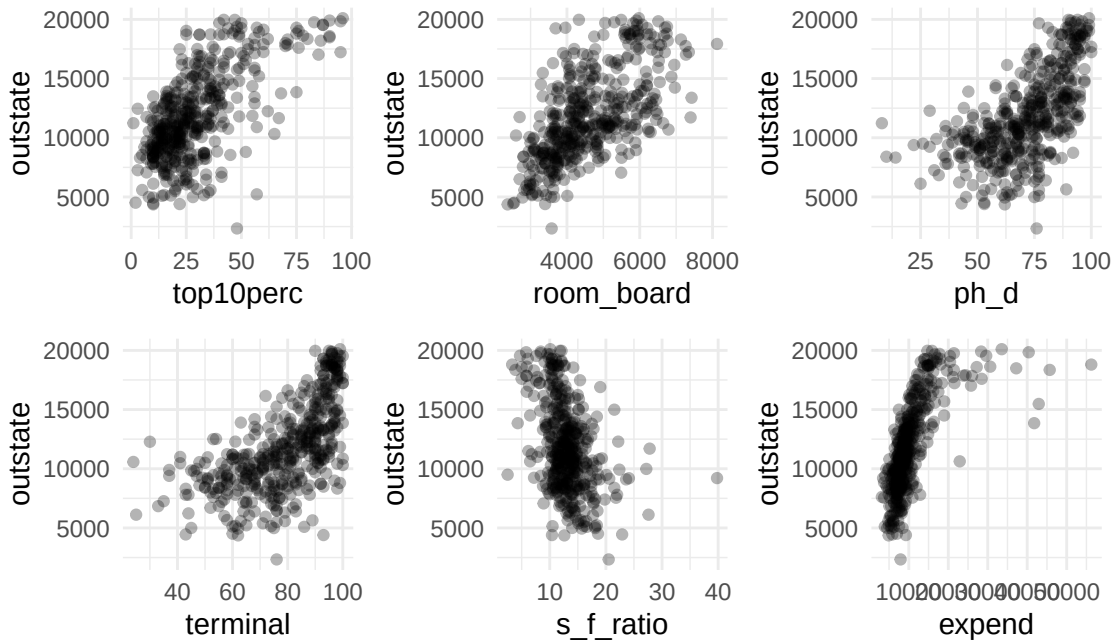
Correlation plot shows that some variables are highly correlated with outstate and there is multicollinearity.

```
corrplot::corrplot(cor(training_df %>% select(where(is.numeric)) %>% relocate(outstate)), method = "cir
```



Plot scatterplot of response variables with selection of highly correlated predictors including top10perc, room_board, ph_d, terminal, s_f_ratio and expend. Only s_f_ratio is negatively correlated.

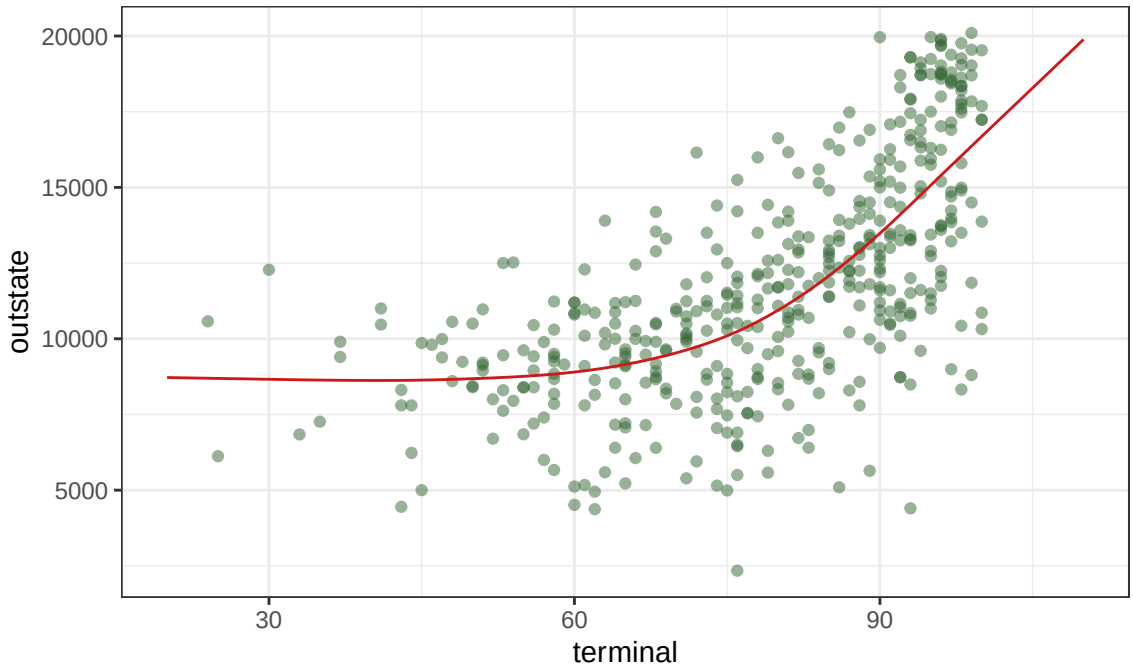
```
library(patchwork)
p1 = ggplot(training_df, aes(x=top10perc, y=outstate)) + geom_point(alpha=0.3)
p2 = ggplot(training_df, aes(x=room_board, y=outstate)) + geom_point(alpha=0.3)
p3 = ggplot(training_df, aes(x=ph_d, y=outstate)) + geom_point(alpha=0.3)
p4 = ggplot(training_df, aes(x=terminal, y=outstate)) + geom_point(alpha=0.3)
p5 = ggplot(training_df, aes(x=s_f_ratio, y=outstate)) + geom_point(alpha=0.3)
p6 = ggplot(training_df, aes(x=expend, y=outstate)) + geom_point(alpha=0.3)

(p1 + p2 + p3)/(p4 + p5 + p6)
```



(b) Fit smoothing spline models using Terminal as the only predictor of Outstate

Fit on train data

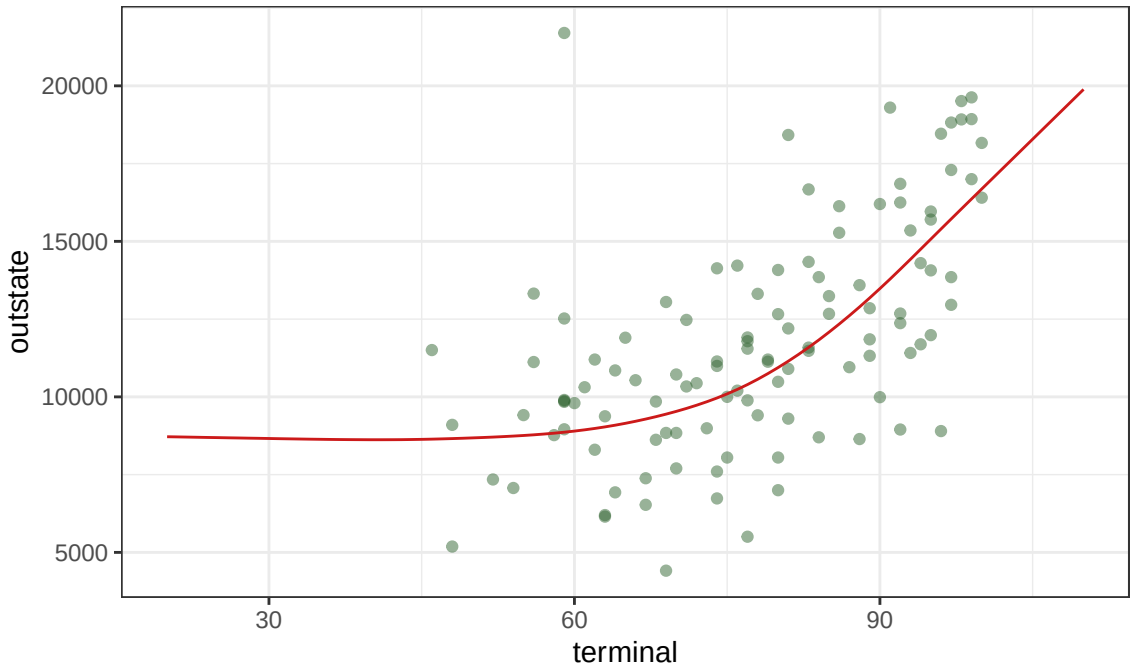
```
set.seed(777)
fit.ss <- smooth.spline(training_df$terminal, training_df$outstate)

terminal.grid <- seq(from = 20, to = 110, by = 1)

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

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

p <- ggplot(data = training_df, 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()
```



Degree of freedom is 4.3635782

Fit on test data

```
set.seed(777)
fit.ss <- smooth.spline(training_df$terminal, training_df$outstate)

terminal.grid <- seq(from = 20, to = 110, by = 1)

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

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

p <- ggplot(data = testing_df, 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()
```



Model captures the trend of outstate on the test set as well.

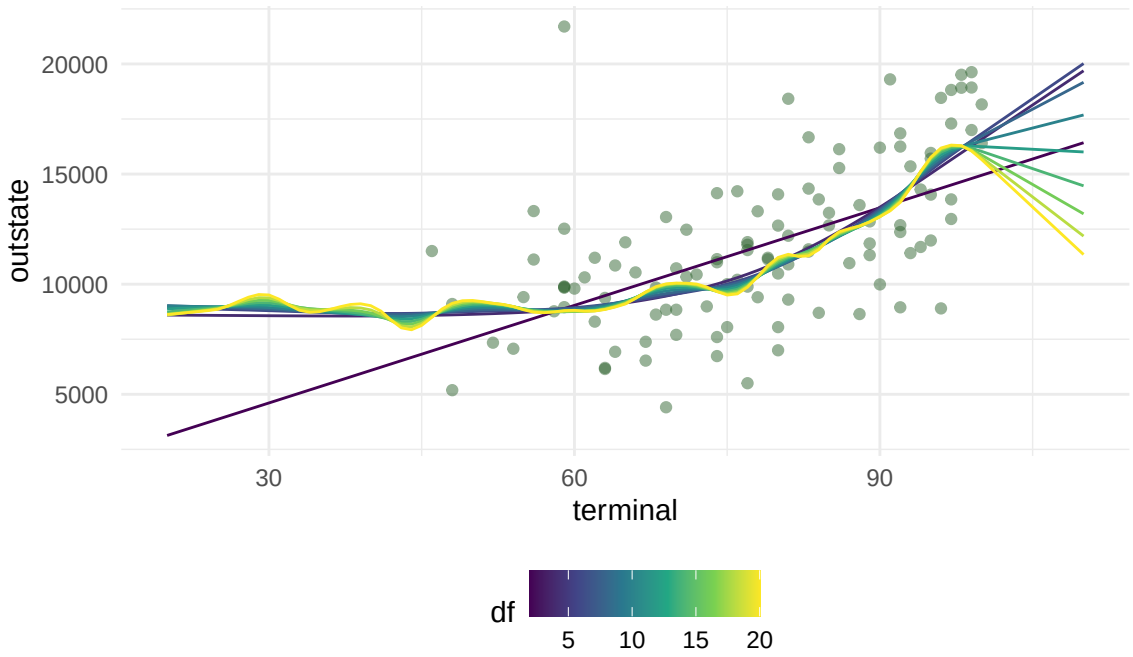
Generalized cross-validation and visualize it

Set df candidates to a sequences from 2 to 20 with step as 2, fit and record the result

```
pred.ss.df <- data.frame(pred = pred.ss$y,
                         x = terminal.grid)

flag = TRUE

for (df in seq(2, 20, by=2)){
  fit.ss <- smooth.spline(training_df$terminal, training_df$outstate, df = df)
  pred.ss <- predict(fit.ss,
                     x = terminal.grid)
  pred.ss.df <- data.frame(pred = pred.ss$y,
                           x = terminal.grid,
                           df = df)
  if (flag){
    pred.ss.df.all = pred.ss.df
    flag = FALSE
  }
  else{
    pred.ss.df.all = rbind(pred.ss.df.all, pred.ss.df)
  }
}
```

Visualize it. The larger the df, the more non-linear fit.

```
p +
geom_line(aes(x = x, y = pred, group = df, color = df), data = pred.ss.df.all)
```



(c) Fit a generalized additive model (GAM) using all the predictors.

Training

```
library(mgcv)
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
## This is mgcv 1.8-36. For overview type 'help("mgcv-package")'.
```

```
outcome = "outstate"
variables = colnames(x_train)

# fully parameterized
f = as.formula(
  paste(outcome,
        paste(variables, collapse = " + "),
        sep = " ~ "))

gam.m1 <- gam(f, data = training_df)
gam.m2 <- gam(outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
```

```
    p_undergrad + room_board + books + personal + ph_d + s(terminal) +
    s_f_ratio + perc_alumni + expend + grad_rate, data = training_df)
gam.m3 <- gam(outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
    p_undergrad + room_board + books + personal + ph_d + s(terminal) +
    te(s_f_ratio, perc_alumni) + expend + grad_rate, data = training_df)

anova(gam.m1, gam.m2, gam.m3, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
##     p_undergrad + room_board + books + personal + ph_d + terminal +
##     s_f_ratio + perc_alumni + expend + grad_rate
## Model 2: outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
##     p_undergrad + room_board + books + personal + ph_d + s(terminal) +
##     s_f_ratio + perc_alumni + expend + grad_rate
## Model 3: outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
##     p_undergrad + room_board + books + personal + ph_d + s(terminal) +
##     te(s_f_ratio, perc_alumni) + expend + grad_rate
##   Resid. Df Resid. Dev     Df  Deviance      F    Pr(>F)
## 1    436.00 1577792512
## 2    432.79 1527728007  3.209  50064505 5.0774  0.001409 **
## 3    418.98 1296186637 13.815 231541370 5.4547 1.478e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fails to reject they have the same deviance.

Model result for the best GAM model: model3

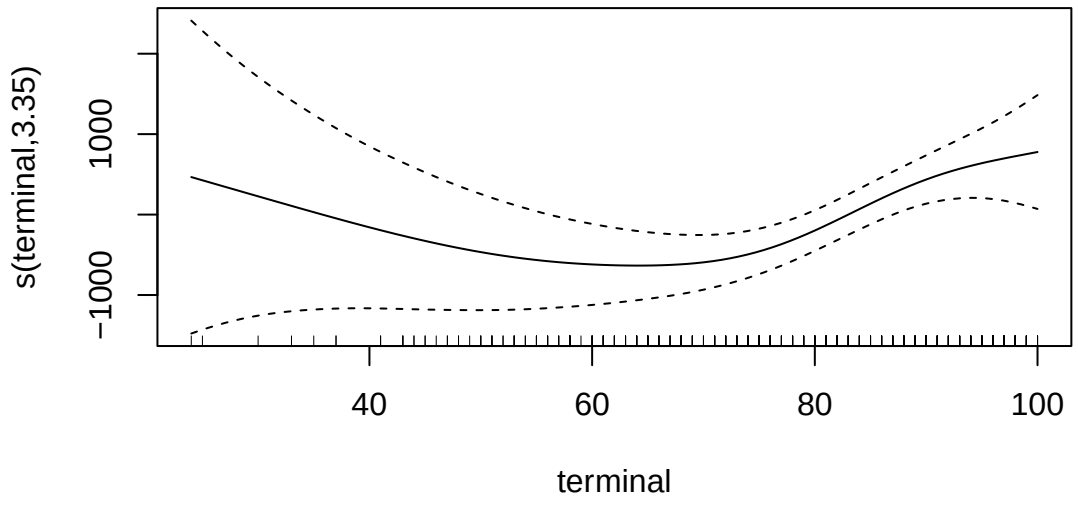
```
summary(gam.m3)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## outstate ~ apps + accept + enroll + top10perc + top25perc + f_undergrad +
##     p_undergrad + room_board + books + personal + ph_d + s(terminal) +
##     te(s_f_ratio, perc_alumni) + expend + grad_rate
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2543.26109  971.91399   2.617 0.009196 **
## apps           0.05417    0.10744   0.504 0.614419
## accept         1.09001    0.19560   5.573 4.48e-08 ***
## enroll        -3.27164    0.85080  -3.845 0.000139 ***
## top10perc     28.56125   14.41426   1.981 0.048189 *
## top25perc     -1.58500   11.22486  -0.141 0.887776
## f_undergrad    0.02982    0.12922   0.231 0.817604
## p_undergrad   -0.22005    0.13161  -1.672 0.095269 .
## room_board     0.92512    0.09924   9.322  < 2e-16 ***
## books         -0.52390    0.50221  -1.043 0.297459
```

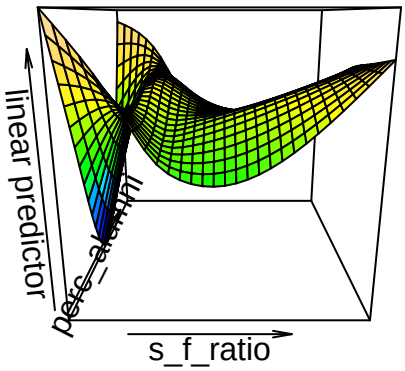
```
## personal      -0.37659    0.15121  -2.490 0.013141 *
## ph_d          23.48824    9.95829   2.359 0.018796 *
## expend         0.21436    0.03014   7.113 4.91e-12 ***
## grad_rate     15.76908    6.51363   2.421 0.015902 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                               edf Ref.df     F p-value
## s(terminal)                 2.451  3.112 3.086  0.0246 *
## te(s_f_ratio,perc_alumni) 14.701 16.912 5.993  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.779   Deviance explained = 79.4%
## GCV = 3.2995e+06  Scale est. = 3.0726e+06  n = 453
```

Visualization

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




```
vis.gam(gam.m3, view = c("s_f_ratio","perc_alumni"),
color = "topo")
```



###

Prediction on the test set

```
pred_y = predict.gam(gam.m3, newdata = as.tibble(x_test))
```

```
## Warning: `as.tibble()` was deprecated in tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
```

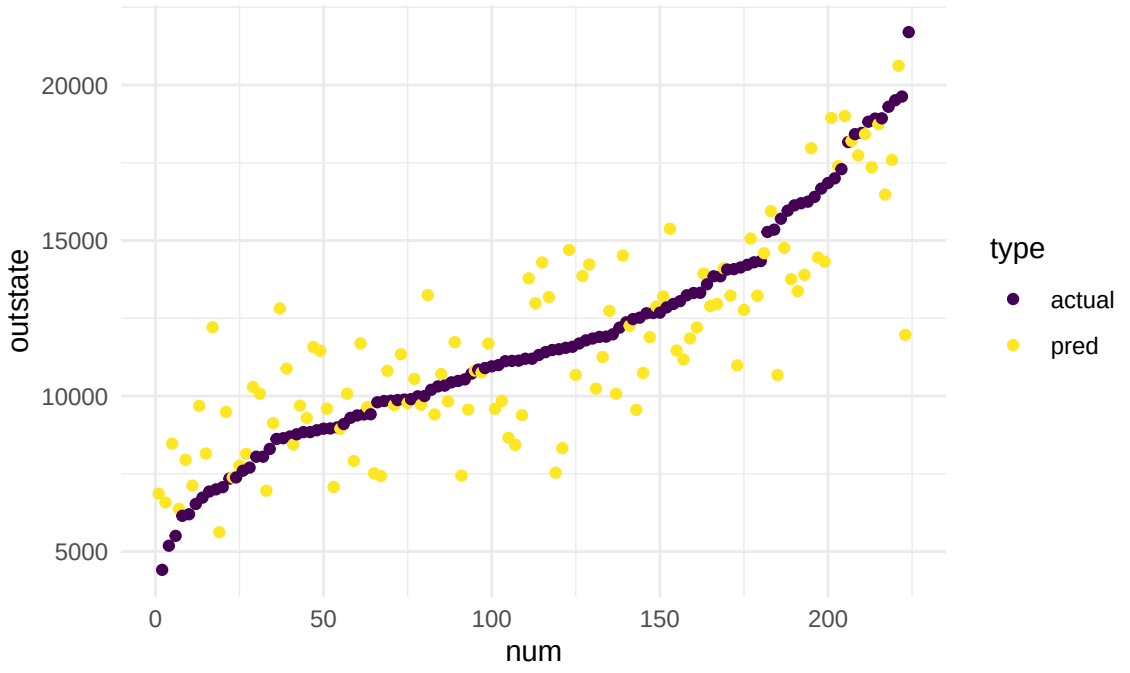
```
actual_y = y_test

# Metrics
rmse = sqrt(mean((pred_y - actual_y)^2)) # Calculate test MSE
mae = mean(abs(pred_y - actual_y))
```

Final RMSE is 2079.5996225, MAE is 1592.099841, while the median in the test set is 1.126×10^4. The predicton error is acceptable.

We can also see from the prediction compare plot that the model fits well.

```
y_compare_df =
  tibble(pred = pred_y,
         actual = actual_y) %>%
  arrange(actual) %>%
    pivot_longer(
      cols = pred:actual,
      names_to = 'type',
      values_to = 'outstate'
    )

y_compare_df %>%
  mutate(
```

```
    num = seq(1, nrow(y_compare_df)),
    type = as.factor(type)
        ) %>%
  ggplot(aes(x = num, y = outstate), color = type) +
  geom_point(aes(colour = type)) +
  theme(legend.position="right")
```



(d) Train a multivariate adaptive regression spline (MARS) model using all the predictors

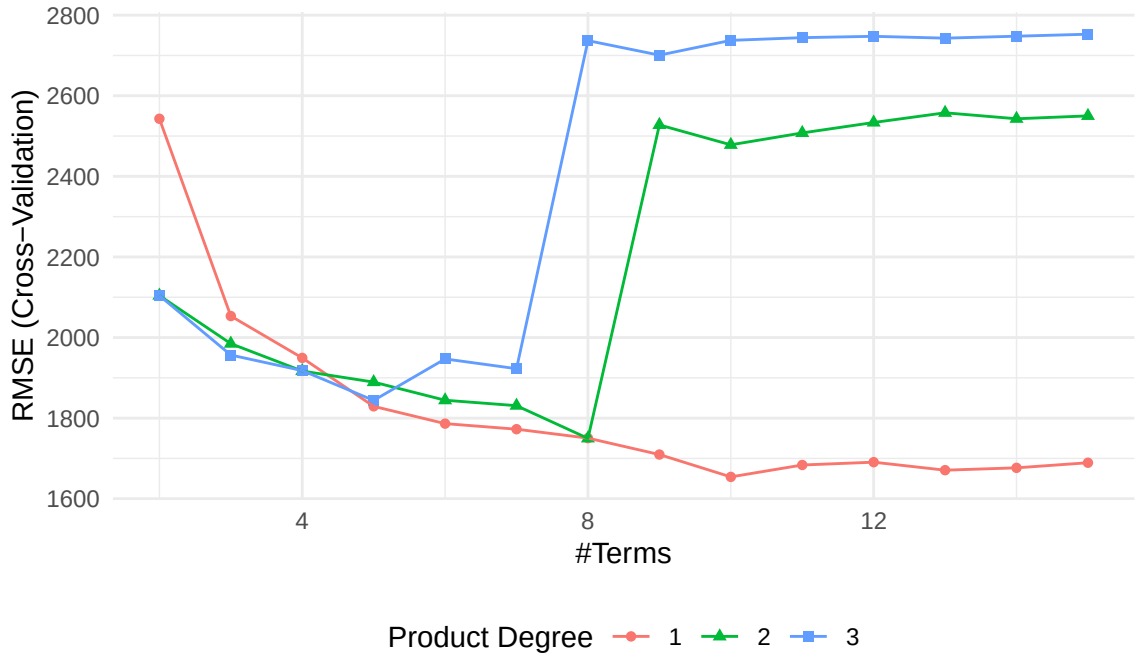
Training MARS

```
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:15)
ctrl1 <- trainControl(method = "cv", number = 10)
mars.fit <- train(x_train, y_train,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)
```

```
## Loading required package: earth
```

```
## Warning: package 'earth' was built under R version 4.1.2
```

```
ggplot(mars.fit)
```




```
# parameters for optimized result
mars.fit$bestTune
```

```
##   nprune degree
## 9     10      1
```

```
# Final model
summary(mars.fit$finalModel)
```

```
## Call: earth(x=matrix[453,16], y=c(7440,12280,11...), keepxy=TRUE, degree=1,
##             nprune=10)
##
##                      coefficients
## (Intercept)           15341.9037
## h(apps-1910)              0.4314
## h(1580-accept)           -1.9431
## h(913-enroll)             5.1327
## h(enroll-913)            -2.3810
## h(1433-f_undergrad)      -1.7917
## h(4440-room_board)       -1.1082
## h(room_board-4440)        0.5022
## h(22-perc_alumni)      -101.2021
## h(15736-expend)           -0.6782
##
## Selected 10 of 21 terms, and 7 of 16 predictors (nprune=10)
## Termination condition: RSq changed by less than 0.001 at 21 terms
## Importance: expend, room_board, perc_alumni, f_undergrad, enroll, apps, ...
## Number of terms at each degree of interaction: 1 9 (additive model)
## GCV 2759994    RSS 1147597147    GRSq 0.8020649    RSq 0.8175158
```

partial dependence

partial dependence between grad_rate and enroll

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

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

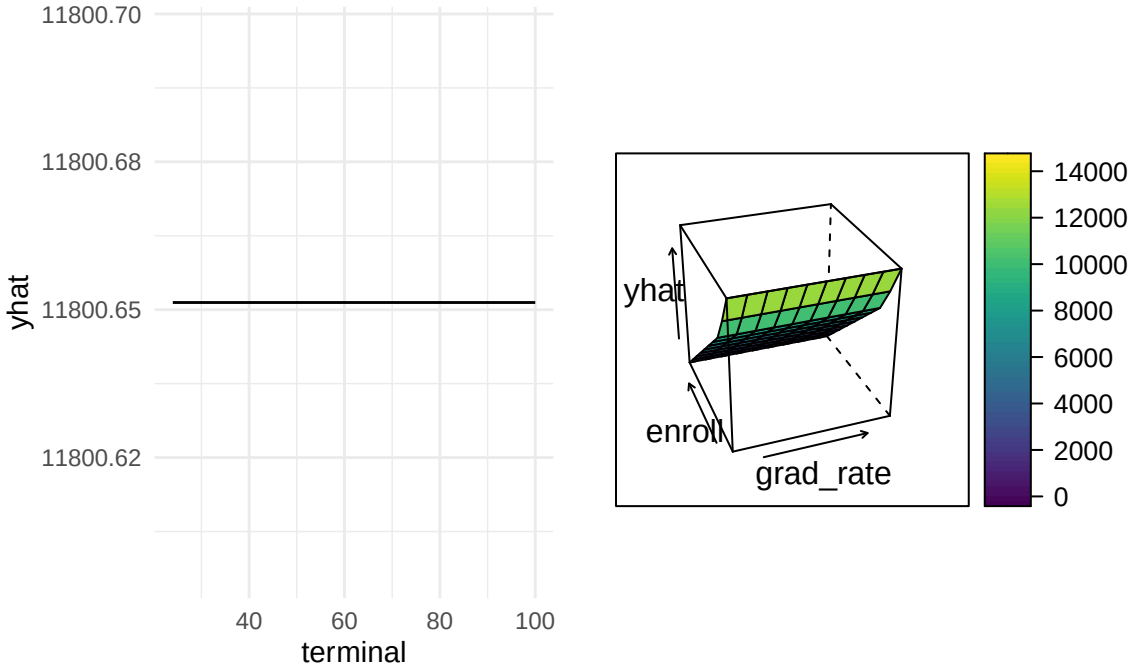
```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

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

```
## Warning: Use of `object[[1L]]` is discouraged. Use `.data[[1L]]` instead.
```

```
## Warning: Use of `object[["yhat"]]` is discouraged. Use `.data[["yhat"]]`
## instead.
```



###

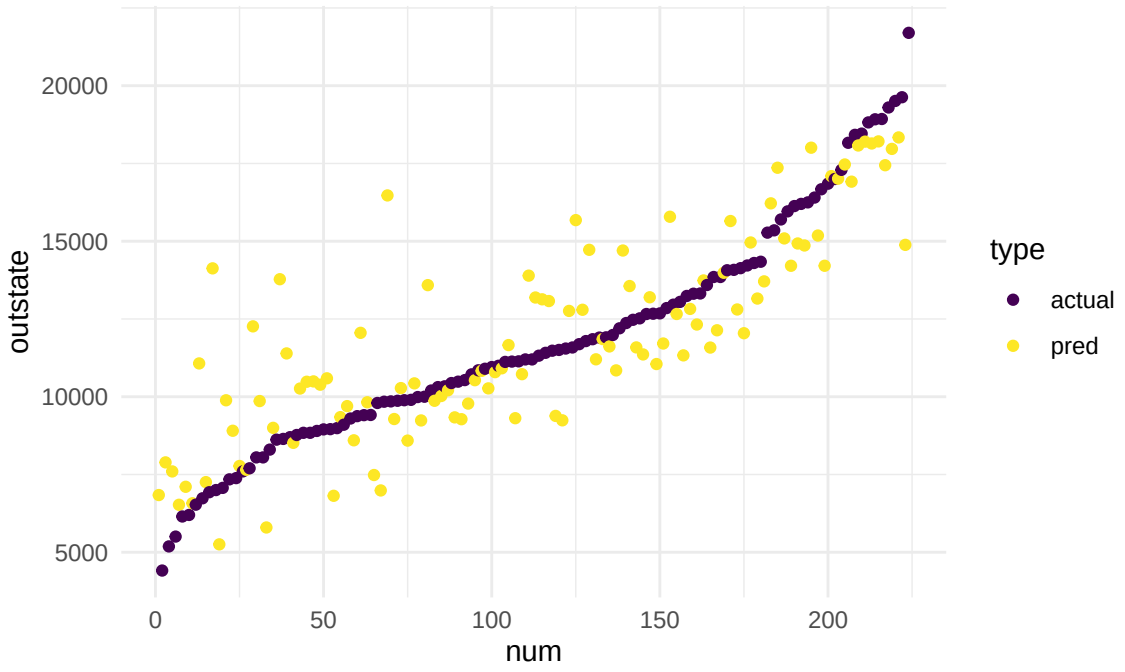
Prediction on the test set

```
pred_y = predict(mars.fit, newdata = as.tibble(x_test))
actual_y = y_test

# Metrics
rmse = sqrt(mean((pred_y - actual_y)^2)) # Calculate test MSE
mae = mean(abs(pred_y - actual_y))
```

Final RMSE is 2007.3293534, MAE is 1495.5303102, while the median in the test set is 1.126×10^4. The predicton error is acceptable.

We can also see from the prediction compare plot that the model fits well.

```
y_compare_df =
  tibble(pred = pred_y,
         actual = actual_y) %>%
  arrange(actual) %>%
    pivot_longer(
      cols = pred:actual,
      names_to = 'type',
      values_to = 'outstate'
    )

y_compare_df %>%
  mutate(
    num = seq(1, nrow(y_compare_df)),
    type = as.factor(type)
         ) %>%
  ggplot(aes(x = num, y = outstate), color = type) +
  geom_point(aes(colour = type)) +
  theme(legend.position="right")
```



(e) Model selection