Homework 2

Tucker Morgan - tlm2152

3/8/2022

```
library(tidyverse)
library(caret)
```

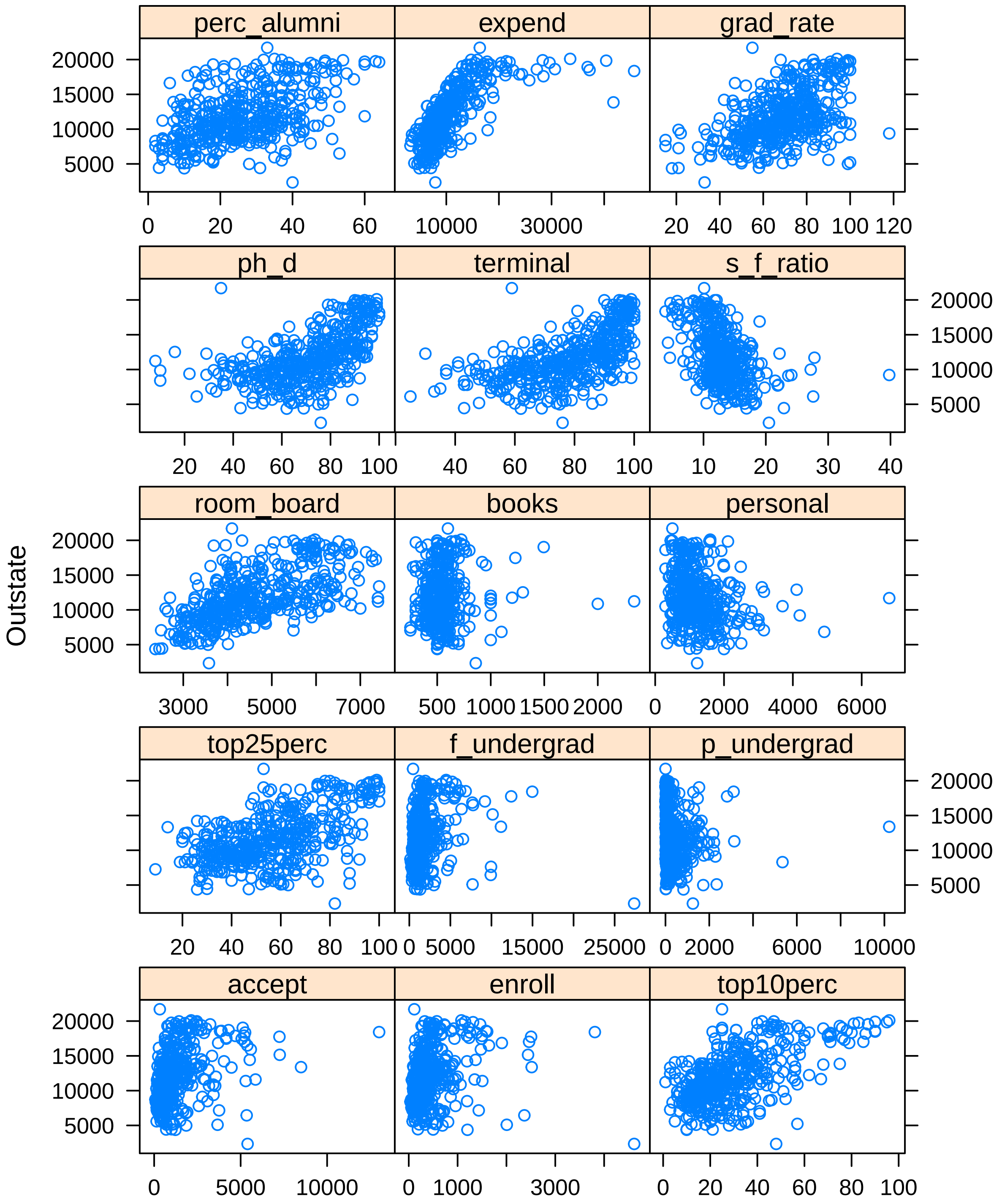
First, let's import and partition our data set.

```
set.seed(100)
college_full <- read_csv("./Homework 2/College.csv") %>%
  janitor::clean_names()
# creating data partitioning index
part_index <- createDataPartition(y = college_full$outstate,
                                  p = 0.8,
                                  list = FALSE)

college_trn <- college_full[part_index, ] %>% # training data
  select(-college) # removing college name because unique identifier
college_tst <- college_full[-part_index, ] %>% # test data
  select(-college) # removing college name because unique identifier
```

Part a

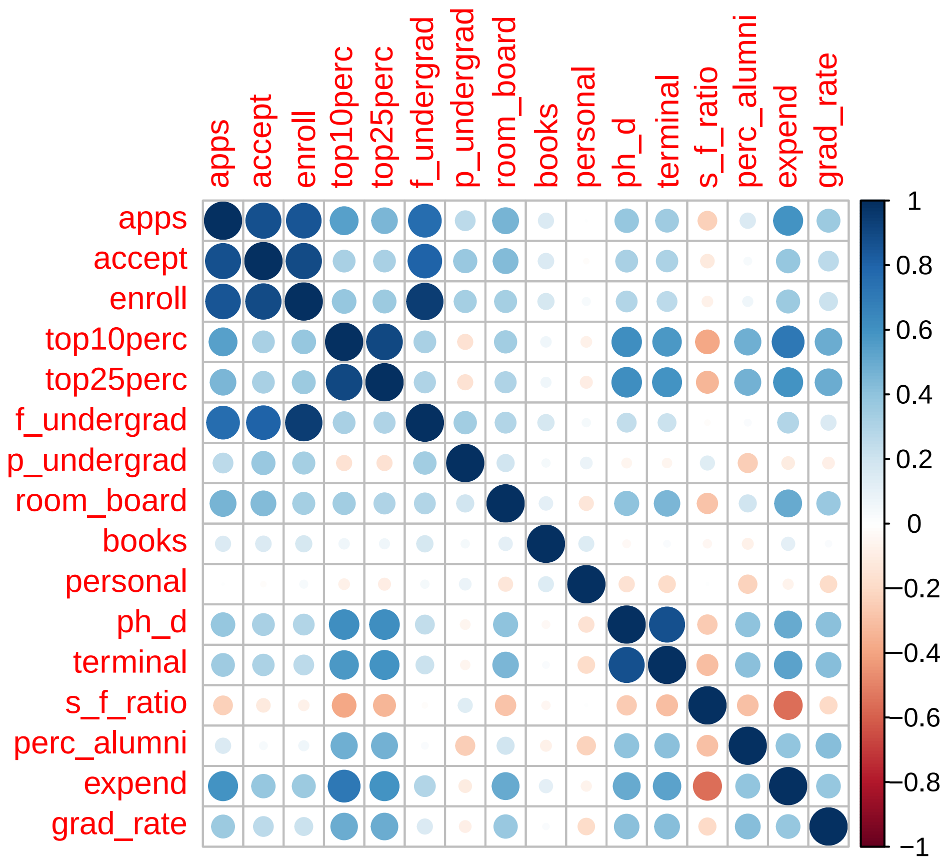
And now we'll look at some exploratory analysis using the training data set.

```
# making a model matrix for caret
college_x <- model.matrix(outstate ~ ., college_trn)[,-1]
college_y <- college_trn$outstate

# setting plot parameters
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.1, .1, .6, .4)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, .8)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)

featurePlot(college_x[,-1], college_y,
            plot = "scatter",
            labels = c("", "Outstate"),
            type = c("p"),
            layout = c(3, 5))
```



Looking at this plot, there are varying relationships between the response, `outstate`, and the array of predictors. It seems like there could be linear relationships between out of state tuition and room and board costs (`room_board`), percentage of new students from the top 10% of their class (`top10perc`), and percentage of new students from the top 25% of their class (`top25perc`). Let's look at a a correlation plot of these predictors.

```
corrplot::corrplot(cor(college_x),
                   method = "circle",
                   type = "full")
```



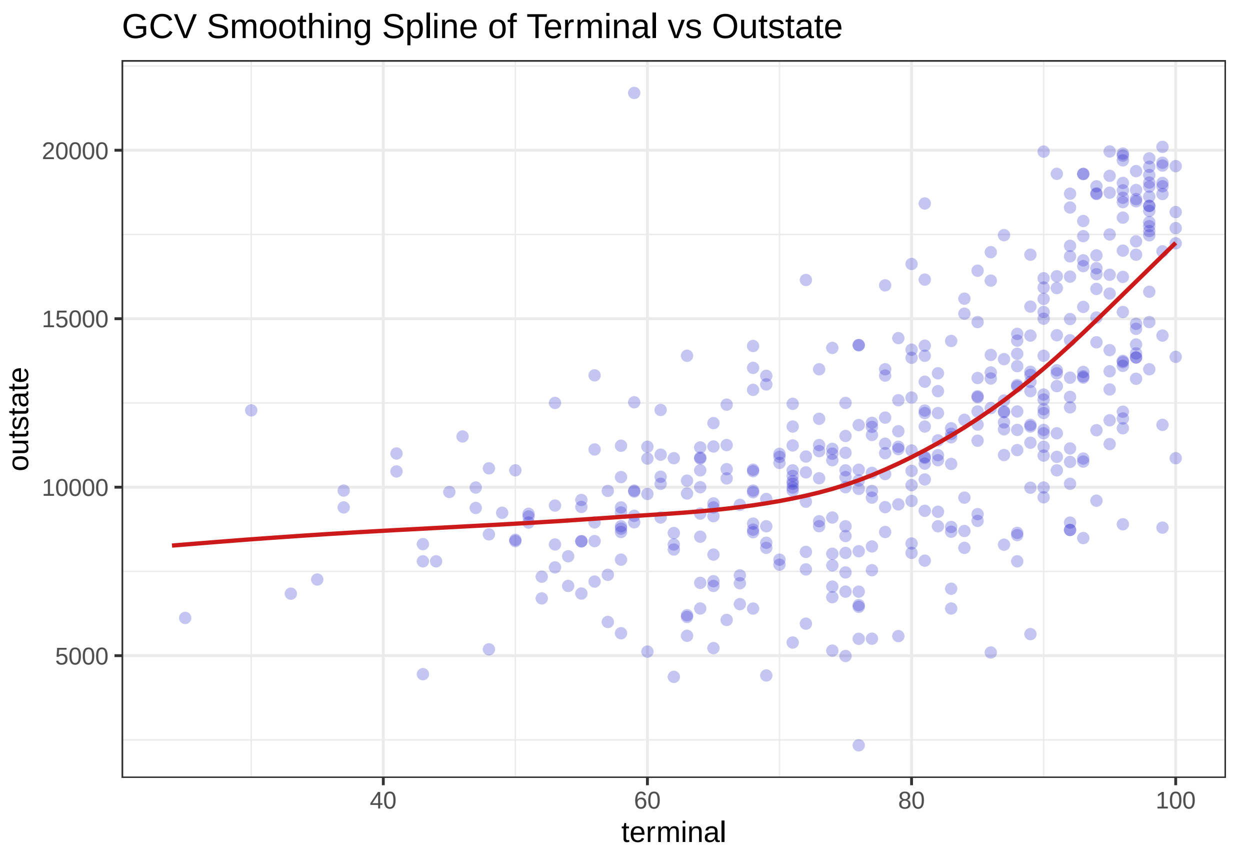
It's clear from this plot that the number of applications, number of accepted students, and number of enrolled students have a high correlation with each other along with the number of full time undergraduate students, `f_undergrad`. There is also an understandably strong relationship between `top10perc` and `top25perc`. Another strong correlation is between percent of faculty with PhD's and percent of faculty with terminal degrees. This is likely because terminal degrees are often PhD's.

Part b

Next, I'll fit smoothing spline models using `terminal` as the only predictor of `outstate` with varying degrees of freedom.

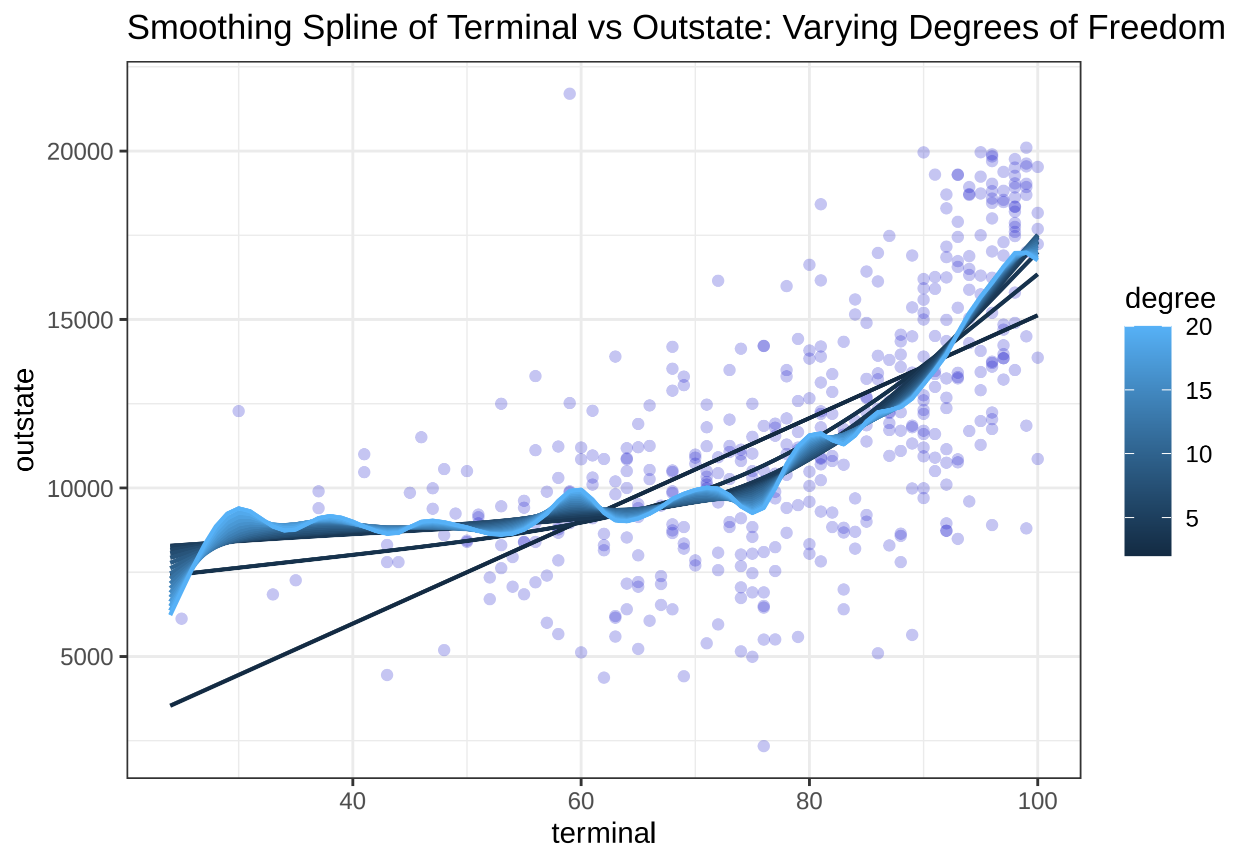
```
college_ss <- smooth.spline(x = college_trn$terminal,
                            y = college_trn$outstate)
# creating grid of values over range of terminal
terminal_grid <- seq(from = min(college_full$terminal),
                     to = max(college_full$terminal),
                     by = 1)
# predictions based on terminal grid
pred_ss_opt <- predict(college_ss,
                       x = terminal_grid)
```

```r
pred_ss_df <- data.frame(pred = pred_ss_opt$y,
                         terminal = terminal_grid)

ggplot(aes(x = terminal, y = outstate), data = college_trn) +
  geom_point(color = rgb(.1, .1, .8, .25)) +
  geom_line(aes(x = terminal, y = pred),
            color = rgb(.8, .1, .1, 1),
            size = 0.75,
            data = pred_ss_df) +
  theme_bw() +
  labs(title = "GCV Smoothing Spline of Terminal vs Outstate")
```



The plot above shows the smoothing spline model obtained when using the degree of freedom obtained from generalized cross-validation (GCV), which is equal to 4.735. Now let's look at plots for a few other degrees of freedom.

```r
# function to create the smooth spline predictions
smoothie <- function(freedom){
  ss_fit <- smooth.spline(x = college_trn$terminal,
                          y = college_trn$outstate,
                          df = freedom)
  pred_ss <- predict(ss_fit,
                     x = terminal_grid)
  pred_ss_df_freedom <- data.frame(pred = pred_ss,
                                   terminal = terminal_grid,
                                   degree = freedom)
  return(pred_ss_df_freedom)
}
# creating a list to store for loop results
spline_list <- list()
# iterates over desired degrees of freedom
for (i in 2:20){
  output <- smoothie(i)
  spline_list[[i]] <- output
}
# converting list to data frame for ggplot
spline_df <- plyr::ldply(spline_list, data.frame) %>%
  group_by(degree)


ggplot(aes(x = terminal, y = outstate), data = college_trn) +
  geom_point(color = rgb(.1, .1, .8, .25)) +
  geom_line(aes(x = terminal, y = pred.y, group = degree, color = degree),
            size = 0.75,
            data = spline_df) +
  theme_bw() +
  labs(title = "Smoothing Spline of Terminal vs Outstate: Varying Degrees of Freedom")
```



The plot of the GCV-optimized degree of freedom seems to fit the data reasonably well. It has some flexibility and follows the pattern of the data, but it does not seem to overfit or contort in odd ways. In the second plot of the varying degrees of freedom, we see the smoothing spline curves become more flexible as degree of freedom increases. The darker lines are either linear or have slight flexibility, however the lighter blue curves gain increased flexibility and show higher variance across the data. The model with the highest degrees of freedom, 20, seems to swing rather dramatically at different points in the data.

Part c

Next, let's look at a generalized additive model (GAM) using all predictors.

```
set.seed(100)
# here we will use caret
ctrl1 <- trainControl(method = "cv", number = 10)

college_gam <- train(x = college_x,
                     y = college_y,
                     method = "gam",
                     tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE, FALSE)),
                     trControl = ctrl1)
college_gam$bestTune
```

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

```
college_gam$finalModel
```

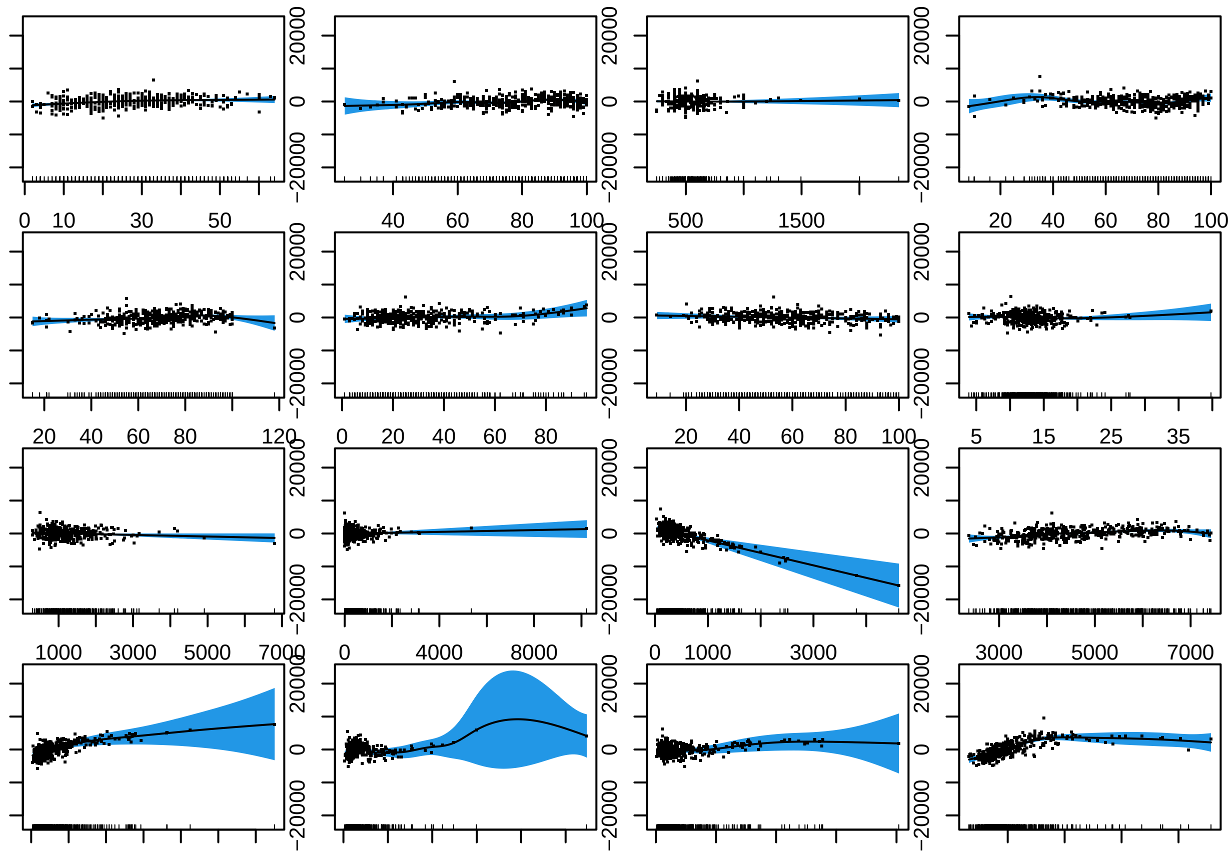
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(books) + s(ph_d) +
##     s(grad_rate) + s(top10perc) + s(top25perc) + s(s_f_ratio) +
##     s(personal) + s(p_undergrad) + s(enroll) + s(room_board) +
##     s(accept) + s(f_undergrad) + s(apps) + s(expend)
##
## Estimated degrees of freedom:
## 2.17 5.25 1.44 6.55 4.52 4.08 1.00
## 2.59 1.00 1.00 1.00 5.27 3.17 7.49
## 3.60 5.16  total = 56.29
##
## GCV score: 2608418
```

```
summary(college_gam)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ s(perc_alumni) + s(terminal) + s(books) + s(ph_d) +
##     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)
```

```
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11826.64      71.01   166.5   <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) 2.166  2.750  7.260 0.000275 ***
## s(terminal)    5.252  6.371  2.055 0.059172 .
## s(books)       1.443  1.752  0.213 0.815758
## s(ph_d)        6.547  7.577  3.429 0.001209 **
## s(grad_rate)   4.516  5.562  4.657 0.000236 ***
## s(top10perc)   4.084  5.094  1.196 0.300785
## s(top25perc)   1.000  1.000  1.217 0.270601
## s(s_f_ratio)   2.586  3.299  1.065 0.286169
## s(personal)    1.000  1.000  3.748 0.053580 .
## s(p_undergrad) 1.000  1.000  0.994 0.319410
## s(enroll)      1.000  1.000 22.454 3.29e-06 ***
## s(room_board)  5.270  6.421  6.256 2.15e-06 ***
## s(accept)      3.170  3.990  3.538 0.007575 **
## s(f_undergrad) 7.487  8.222  4.482 2.16e-05 ***
## s(apps)        3.602  4.431  1.660 0.146330
## s(expend)      5.163  6.317 23.043  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.837   Deviance explained = 85.7%
## GCV = 2.6084e+06  Scale est. = 2.2843e+06  n = 453
```

```
par(mar = c(1, 1, 1, 1))
par(mfrow = c(4, 4))
plot(college_gam$finalModel, residuals = TRUE, all.terms = TRUE, shade = TRUE, shade.col = 4)
```



In our final model, each predictor was considered for nonlinearity as shown by the `s(predictor)` output from `college_gam$finalModel`. In `caret`, any predictor with more than 10 unique data points is considered for nonlinearity. However, if a predictor has fewer than 10 unique values, it is not considered and some flexibility is lost. Here, this is thankfully not a concern.

When looking at the model summary, we see an output of "parametric coefficients" and "smooth terms". In this model, only the intercept is included in the parametric coefficients section, which indicates linear terms. This makes sense because all of our predictors were considered for non-linearity. Looking at the "edf" or estimated degrees of freedom for our smooth terms, we see several instances of edf = 1.00. This indicates a linear relationship. So although nonlinear terms were considered for `top25perc`, `personal`, `p_undergrad`, and `enroll`, ultimately these terms show a linear relationship with the outcome in this model. We can see that the other predictors all have degrees of freedom greater than 1, indicating nonlinear terms. Now let's look at the test error.

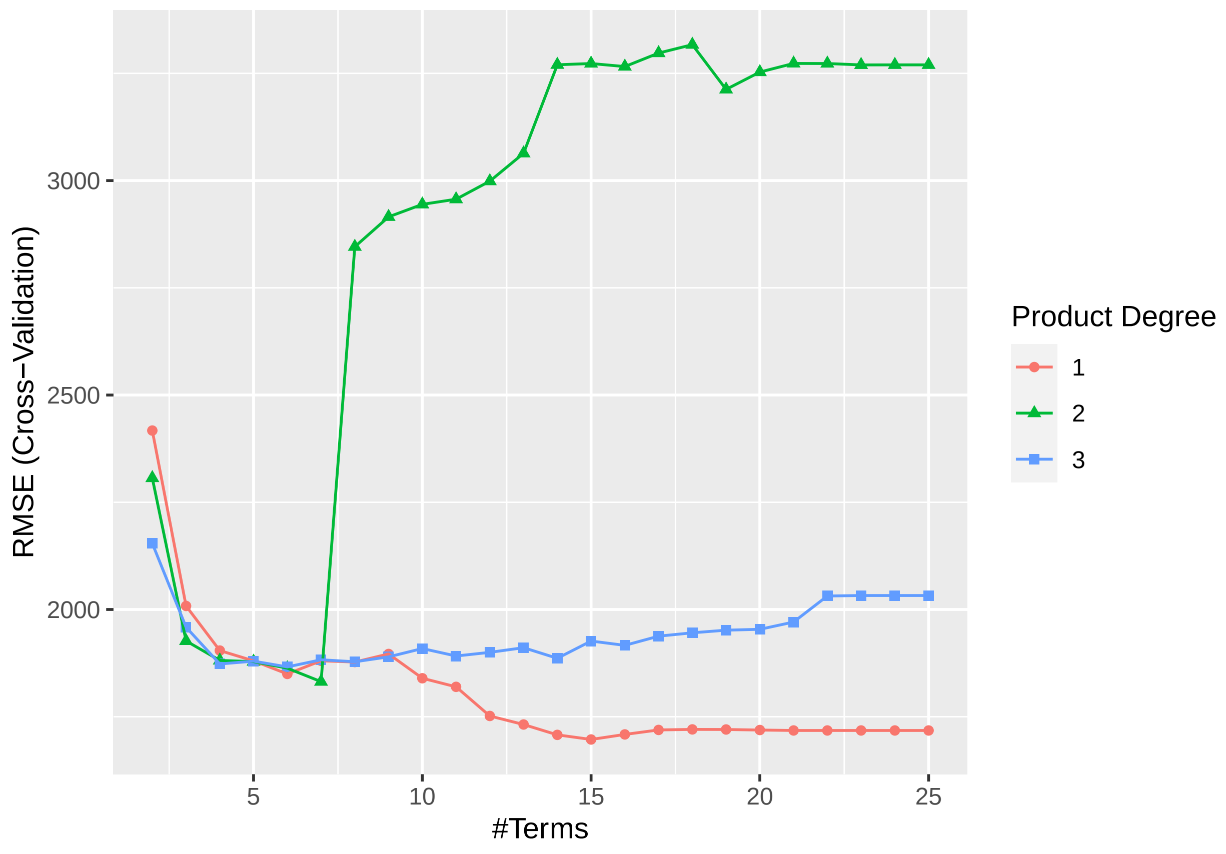
```
gam_predict <- predict(college_gam,
                       newdata = college_tst)
gam_error <- postResample(pred = gam_predict,
                          obs = college_tst$outstate)
```

We find a test RMSE of 2023.31 using this GAM model on the test data set.

Part d

Let's next look at a multivariate adaptive regression spline (MARS) model using all of the predictors.

```
set.seed(100)
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:25)
college_mars <- train(x = college_x,
                      y = college_y,
                      method = "earth",
                      tuneGrid = mars_grid,
                      trControl = ctrl1)
ggplot(college_mars)
```



```
college_mars$bestTune
```

```
##    nprune degree
## 14     15      1
```

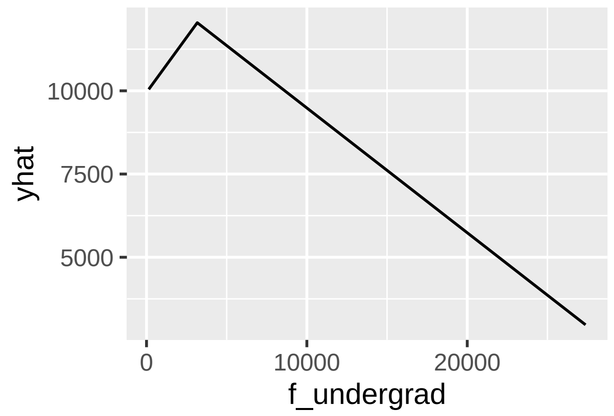
```
coef(college_mars$finalModel) %>% knitr::kable()
```

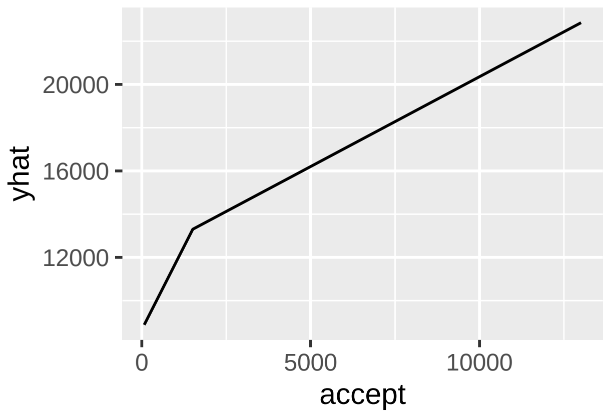
	x
(Intercept)	8273.4621413
h(expend-15687)	-0.6809001
h(grad_rate-83)	-170.2730448
h(5690-room_board)	-0.6828046

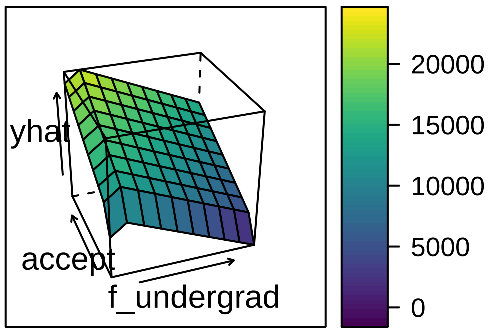
	x
h(f_undergrad-1363)	-0.3747662
h(1363-f_undergrad)	-2.1871511
h(16-perc_alumni)	-140.9339745
h(1445-apps)	1.6898452
h(ph_d-81)	94.6070565
h(903-enroll)	5.1339318
h(accept-1427)	0.8307830
h(1427-accept)	-3.2120520
h(grad_rate-70)	109.4269731
h(expend-5480)	0.6521501
h(top10perc-68)	136.3709747

Reviewing the output above, we can see that the lowest RMSE in the plot is achieved where the product degree is equal to 1 and the number of terms is equal to 15. We see this verified by the `bestTune` object from our fitted model. Looking at the coefficients, we see the predictors found to be predictive of `outstate`. Note the numerical values in the coefficient names. These indicate the location of knots for those predictors. We can look at two predictors specifically, `f_undergrad` and `accept` below.

```
p1 <- pdp::partial(college_mars,
                   pred.var = c("f_undergrad"),
                   grid.resolution = 10) %>% autoplot()
p2 <- pdp::partial(college_mars,
                   pred.var = c("accept"),
                   grid.resolution = 10) %>% autoplot()
p3 <- pdp::partial(college_mars,
                   pred.var = c("f_undergrad", "accept"),
                   grid.resolution = 10) %>%
      pdp::plotPartial(levelplot = FALSE,
                       zlab = "yhat",
                       drape = TRUE,
                       screen = list(z = 20, x = -60))
pdp::grid.arrange(p1, p2, p3, ncol = 2)
```







In this plot, we can see the knots for the two predictors individually and in a hinged plane including both variables. Now, let's look at the test error.

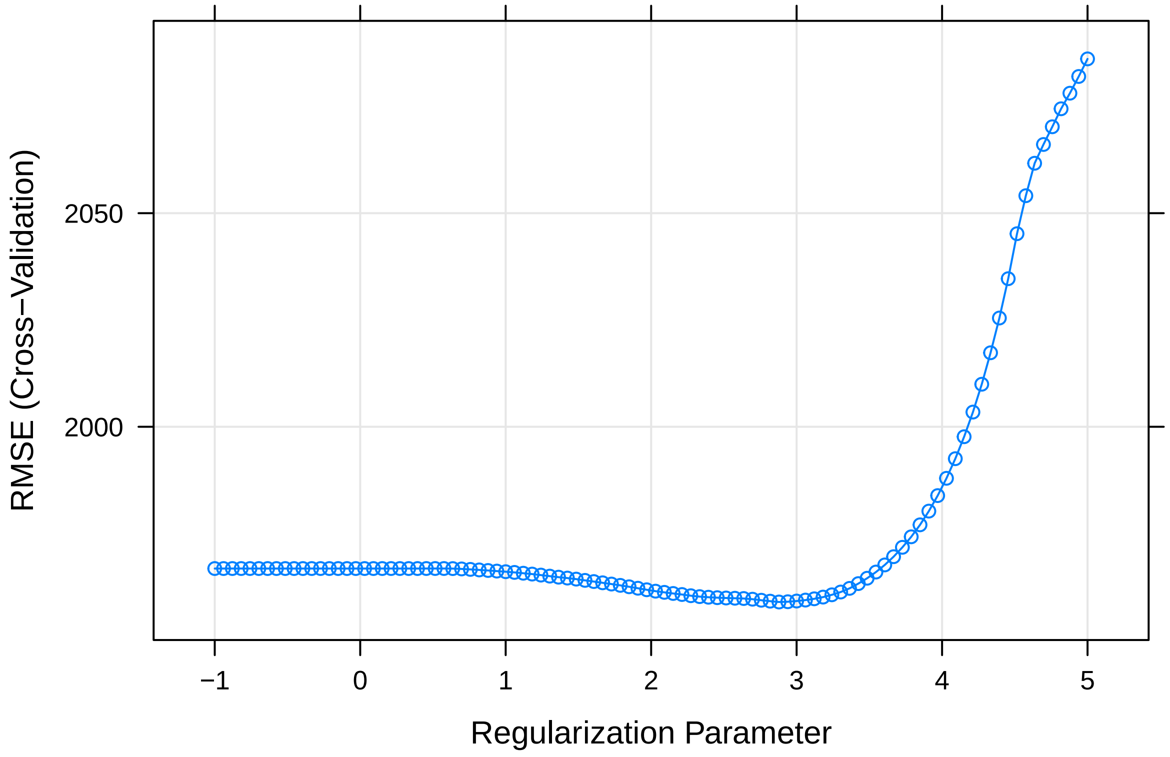
```
mars_predict <- predict(college_mars,
                        newdata = college_tst)
mars_error <- postResample(pred = mars_predict,
                           obs = college_tst$outstate)
```

Using the MARS model, we find a test RMSE of 1973.9.

Part e

We fit a MARS model on the `college` data, but is this better than a linear model for predicting `outstate`? Let's take a look by first fitting a linear model to compare against using LASSO.

```
set.seed(100)

college_lasso <- train(x = college_x,
                       y = college_y,
                       method = "glmnet",
                       tuneGrid = expand.grid(alpha = 1,
                                              lambda = exp(seq(5, -1, length = 100))),
                       trControl = ctrl1,
                       preProcess = c("center", "scale"))
plot(college_lasso, xTrans = log)
```



```
college_lasso$bestTune
```

```
##    alpha   lambda
## 65     1 17.79269
```

```
coef(college_lasso$finalModel, college_lasso$finalModel$lambdaOpt)
```

```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) 11826.637969
## apps             .
## accept       1488.507365
## enroll      -1477.073106
## top10perc     404.144047
## top25perc        .
## f_undergrad      .
## p_undergrad   -66.489063
## room_board    910.768394
## books          15.656775
## personal     -276.132288
## ph_d            1.710696
## terminal      556.674953
## s_f_ratio    -120.660332
## perc_alumni   505.418268
## expend       1153.651033
## grad_rate     339.700381
```

Looking at the output above, we see that the LASSO model removes the following predictors:

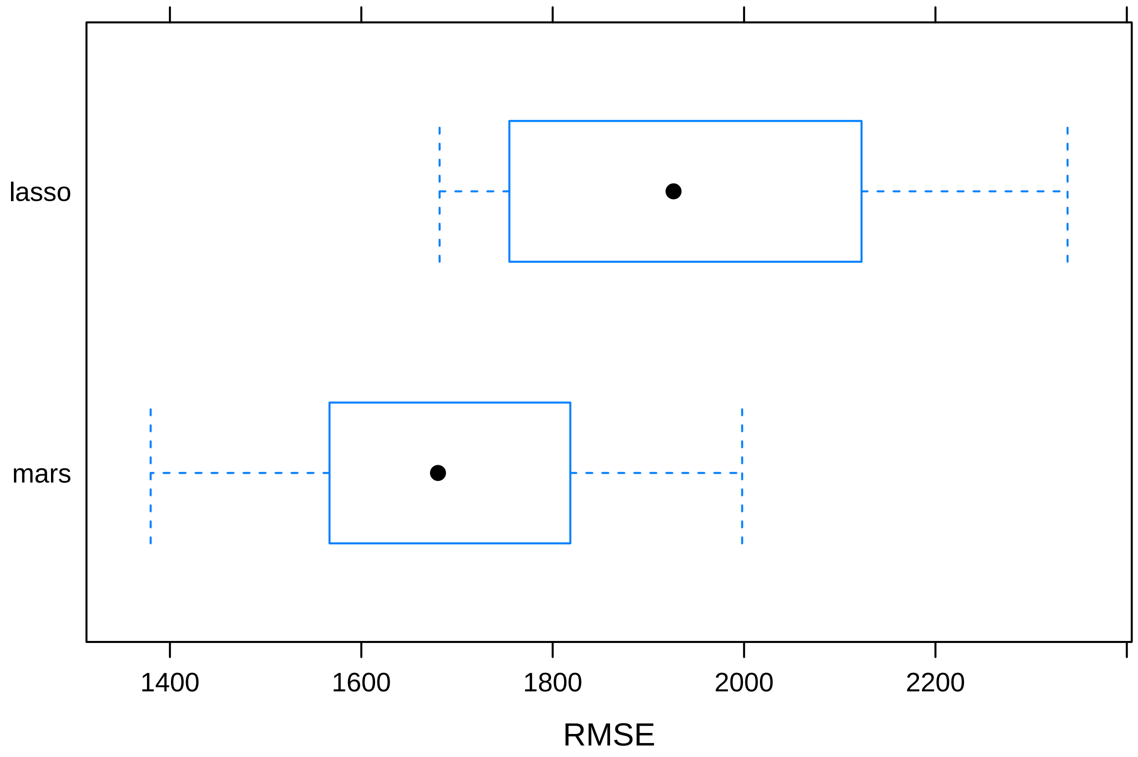
- `apps`;
- `top25perc`;
- `f_undergrad`.

Now, let's compare the cross-validation error of the LASSO model and the MARS model to see which one gives better predictions.

```
set.seed(100)
resamp <- resamples(list(mars = college_mars,
                         lasso = college_lasso))
summary(resamp)
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: mars, lasso
## Number of resamples: 10
##
## MAE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## mars  1112.836 1266.551 1330.984 1330.622 1407.532 1559.823    0
## lasso 1356.678 1437.528 1561.126 1554.287 1656.901 1772.809    0
##
## RMSE
##           Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## mars  1379.862 1569.107 1680.142 1696.957 1811.723 1997.998    0
## lasso 1681.825 1773.708 1926.304 1958.980 2114.387 2338.081    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## mars  0.6635991 0.7714955 0.8132617 0.7929899 0.8278157 0.8888171    0
## lasso 0.6607291 0.6710977 0.7228071 0.7305434 0.7830185 0.8244414    0
```

```
bwplot(resamples(list(mars = college_mars,
                         lasso = college_lasso)),
       metric = "RMSE")
```



Based on the results above, we prefer the MARS model because it has lower mean cross-validation RMSE, and thus it yields better predictions on average. So, we would report a test error of 1973.9.