Nonlinear Models (Splines, GAM, and MARS)

Hana Akbarnejad

3/20/2020

```r
# excluding Columbia University from Data
college_data = read_csv("College.csv") %>% janitor::clean_names() %>%
    filter(college != "Columbia University") %>% select(-college) %>%
    select(outstate, everything())
```

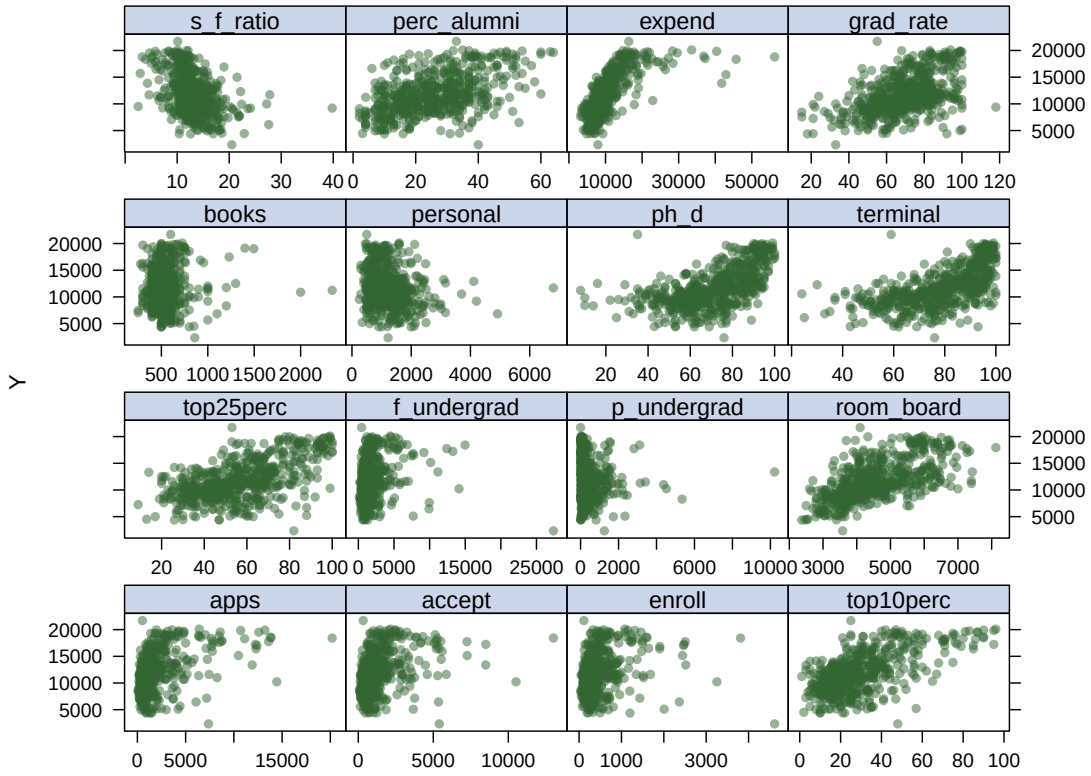
```
## Parsed with column specification:
## cols(
##   College = col_character(),
##   Apps = col_double(),
##   Accept = col_double(),
##   Enroll = col_double(),
##   Top10perc = col_double(),
##   Top25perc = col_double(),
##   F.Undergrad = col_double(),
##   P.Undergrad = col_double(),
##   Outstate = col_double(),
##   Room.Board = col_double(),
##   Books = col_double(),
##   Personal = col_double(),
##   PhD = col_double(),
##   Terminal = col_double(),
##   S.F.Ratio = col_double(),
##   perc.alumni = col_double(),
##   Expend = col_double(),
##   Grad.Rate = col_double()
## )
```

```r
# defining model matrix of response variables and y(outstate
# tuition)
x = model.matrix(outstate ~ ., college_data)[, -1]
y = college_data$outstate
control = trainControl(method = "cv", number = 10)
```

Part a

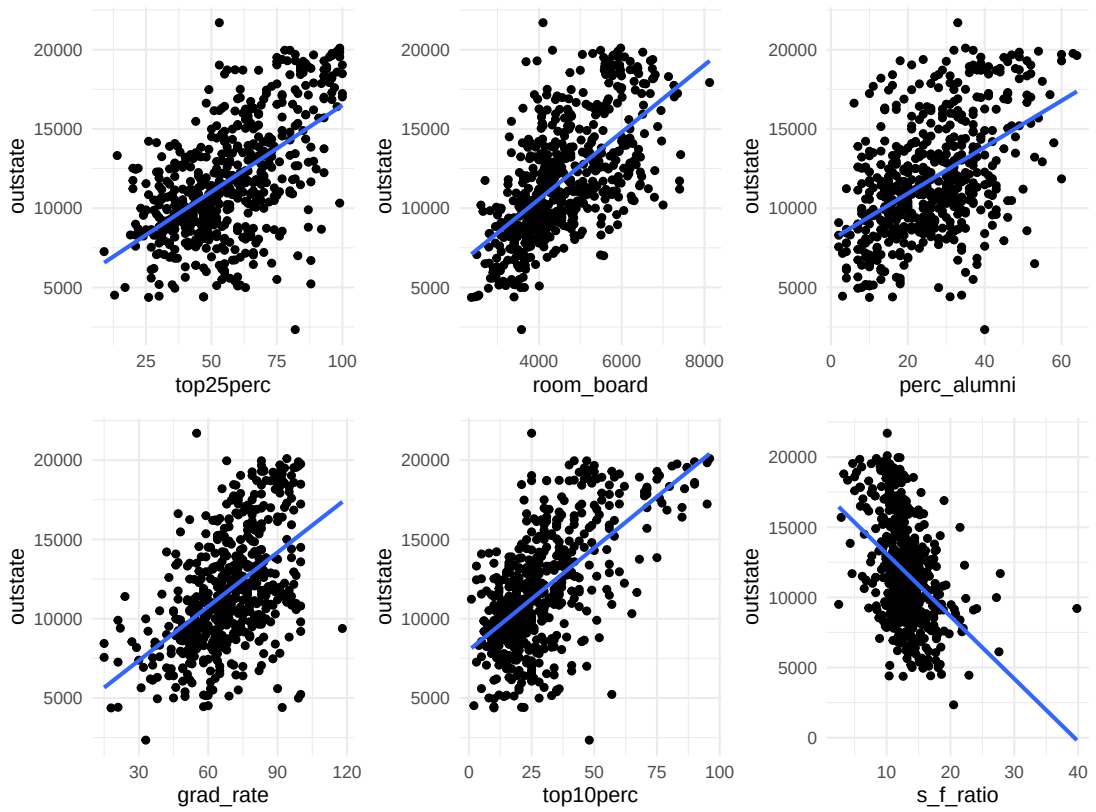
In this part, I will create scatter plots to show the relationship between reponse variable and covariates. I used *featureplot* from *caret package* to do this.

```r
# scatterplot
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(0.2, 0.4, 0.2, 0.5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(0.8, 0.1, 0.1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(0, 0.2, 0.6, 0.2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("", "Y"), type = c("p"),
    layout = c(4, 4))
```



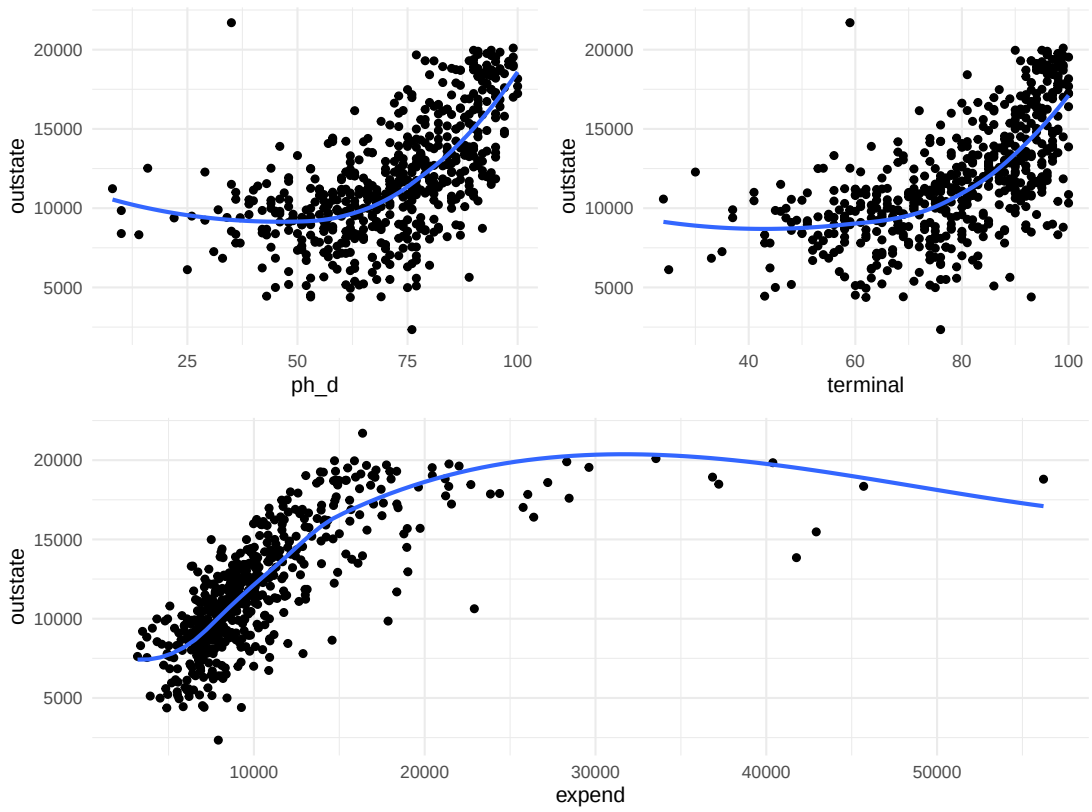
The overall relationship trend between different variables and the response has been depicted above. To be more careful about which variables we pick as non-linear, I am ggoing to draw scatter plots for each variable individually and draw lines on them to see which variables considerably deviate from linearity.

```
l1 = ggplot((college_data), aes(x = top25perc, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l2 = ggplot((college_data), aes(x = room_board, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l3 = ggplot((college_data), aes(x = perc_alumni, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l4 = ggplot((college_data), aes(x = grad_rate, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l5 = ggplot((college_data), aes(x = top10perc, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l16 = ggplot((college_data), aes(x = s_f_ratio, y = outstate)) +
    geom_point() + geom_smooth(method = lm, se = FALSE)
l1 + l2 + l3 + l4 + l5 + l16
```

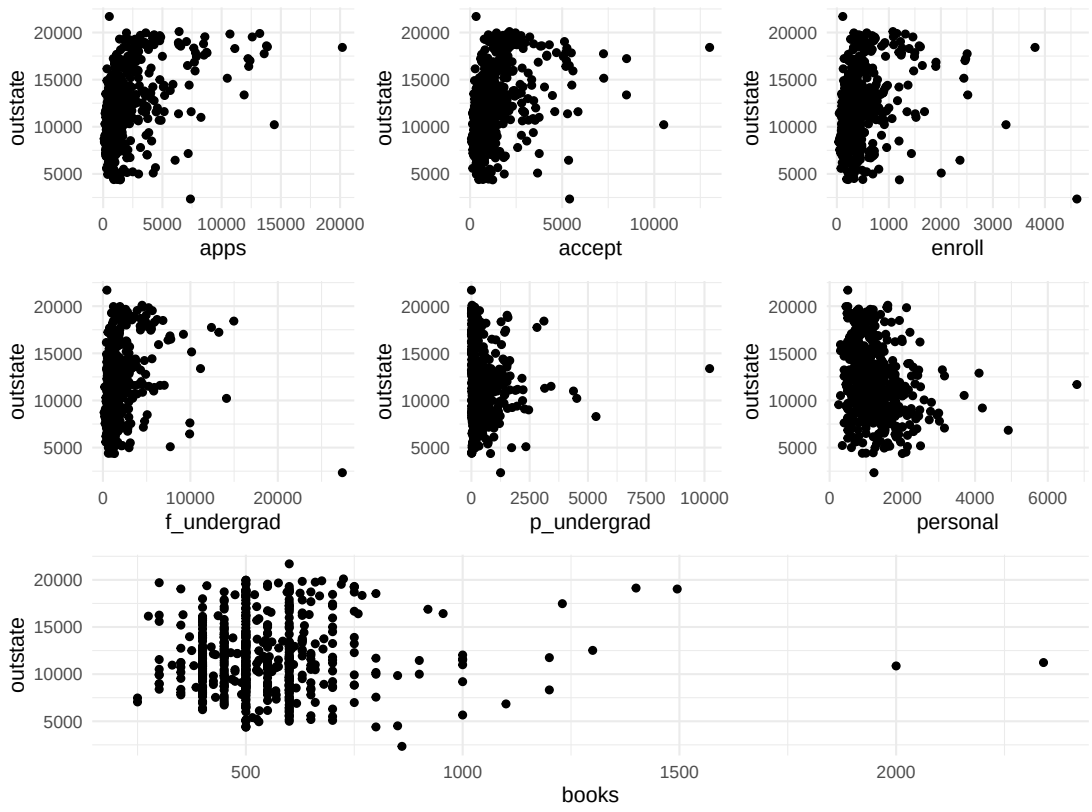



```
l6 = ggplot((college_data), aes(x = ph_d, y = outstate)) + geom_point() +
    geom_smooth(se = FALSE)
l7 = ggplot((college_data), aes(x = terminal, y = outstate)) +
    geom_point() + geom_smooth(se = FALSE)
l8 = ggplot((college_data), aes(x = expend, y = outstate)) +
    geom_point() + geom_smooth(se = FALSE)
(l6 + l7)/l8
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
l9 = ggplot((college_data), aes(x = apps, y = outstate)) + geom_point()
l10 = ggplot((college_data), aes(x = accept, y = outstate)) +
    geom_point()
l11 = ggplot((college_data), aes(x = enroll, y = outstate)) +
    geom_point()
l12 = ggplot((college_data), aes(x = f_undergrad, y = outstate)) +
    geom_point()
l13 = ggplot((college_data), aes(x = p_undergrad, y = outstate)) +
    geom_point()
l14 = ggplot((college_data), aes(x = personal, y = outstate)) +
    geom_point()
l15 = ggplot((college_data), aes(x = books, y = outstate)) +
    geom_point()
(l9 + l10 + l11)/(l12 + l13 + l14)/l15
```



Form scatter plots above, I believe variables *to 10 percent*, *top 25 percent*, *room board*, *alumni percentage*, *s_f_ratio*, and *grad rate* follow a linear trend.

On the other hand, I believe variables *terminal*, *PhD*, and *expend* follow non-linear trends for sure.

For the other variables *apps*, *access*, *enroll*, *f_undergrad*, *p_undergrad*, *personal*, and *books*, however, identifying their trend need more investigation on their outlier values and cannot be determined only by looking at the plots. For the purpose of this homework, I am not going to consider them as non-linear.

Part b

In this part I will fit a **smoothing spline model** using *terminal* as the only predictor and *outstate* as the outcome.

For the degrees of freedom, I will use two methods and plot the resulting fits:

- Using generalized cross-validation (GCV) to obtain degrees of freedom
- Using a range of degrees of freedom

```
terminal_lims = range(college_data$terminal)
terminal_grid = seq(from = terminal_lims[1], to = terminal_lims[2])

# df using GCV
ss_fit = smooth.spline(college_data$terminal, college_data$outstate)
ss_fit
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate)
##
## Smoothing Parameter  spar= 0.8437161  lambda= 0.03935664 (14 iterations)
## Equivalent Degrees of Freedom (Df): 4.468629
## Penalized Criterion (RSS): 323325874
## GCV: 7387023
```

```
ss_pred = predict(ss_fit, x = terminal_grid)
ss_pred_df = data.frame(pred = ss_pred$y, terminal = terminal_grid)

ss_fit$df
```

```
## [1] 4.468629
```

```
# plotting the ss fit
ss_fit2 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 2)
ss_fit2
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 2)
##
## Smoothing Parameter  spar= 1.499938  lambda= 2169.105 (33 iterations)
## Equivalent Degrees of Freedom (Df): 2.000314
## Penalized Criterion (RSS): 988558278
## GCV: 8510179
```

```
ss_pred2 = predict(ss_fit2, x = terminal_grid)
ss_pred_df2 = data.frame(pred = ss_pred2$y, terminal = terminal_grid)

ss_fit3 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 3)
ss_fit3
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 3)
##
## Smoothing Parameter  spar= 0.9677899  lambda= 0.3100395 (11 iterations)
## Equivalent Degrees of Freedom (Df): 3.000349
## Penalized Criterion (RSS): 426499250
## GCV: 7533299
```

```
ss_pred3 = predict(ss_fit3, x = terminal_grid)
ss_pred_df3 = data.frame(pred = ss_pred3$y, terminal = terminal_grid)


ss_fit6 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 6)
ss_fit6
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 6)
##
## Smoothing Parameter  spar= 0.7589078  lambda= 0.009611333 (14 iterations)
## Equivalent Degrees of Freedom (Df): 5.999148
## Penalized Criterion (RSS): 310973774
## GCV: 7405227
```

```
ss_pred6 = predict(ss_fit6, x = terminal_grid)
ss_pred_df6 = data.frame(pred = ss_pred6$y, terminal = terminal_grid)


ss_fit10 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 10)
ss_fit10
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 10)
##
## Smoothing Parameter  spar= 0.6185418  lambda= 0.0009293884 (14 iterations)
## Equivalent Degrees of Freedom (Df): 10.00143
## Penalized Criterion (RSS): 281288576
## GCV: 7458059
```

```
ss_pred10 = predict(ss_fit10, x = terminal_grid)
ss_pred_df10 = data.frame(pred = ss_pred10$y, terminal = terminal_grid)

ss_fit20 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 20)
ss_fit20
```

```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 20)
##
## Smoothing Parameter  spar= 0.4335866  lambda= 4.28497e-05 (12 iterations)
## Equivalent Degrees of Freedom (Df): 20.00251
## Penalized Criterion (RSS): 217712813
## GCV: 7613639
```

```
ss_pred20 = predict(ss_fit20, x = terminal_grid)
ss_pred_df20 = data.frame(pred = ss_pred20$y, terminal = terminal_grid)

ss_fit30 = smooth.spline(college_data$terminal, college_data$outstate,
    df = 30)
ss_fit30
```

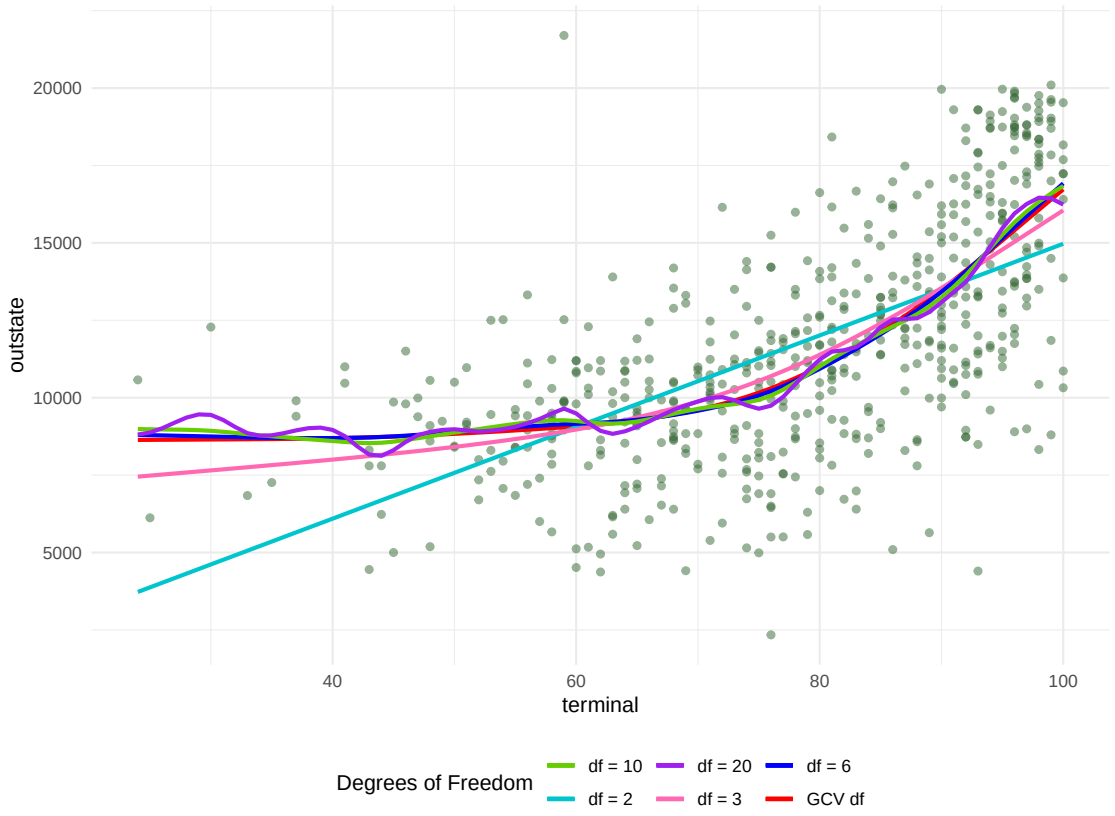
```
## Call:
## smooth.spline(x = college_data$terminal, y = college_data$outstate,
##     df = 30)
```

```
##
## Smoothing Parameter  spar= 0.3207675  lambda= 6.566636e-06 (11 iterations)
## Equivalent Degrees of Freedom (Df): 29.99631
## Penalized Criterion (RSS): 152679983
## GCV: 7772657
```

```
ss_pred30 = predict(ss_fit30, x = terminal_grid)
ss_pred_df30 = data.frame(pred = ss_pred30$y, terminal = terminal_grid)

# plotting
colors = c(`GCV df` = "red", `df = 2` = "turquoise3", `df = 3` = "hotpink",
    `df = 6` = "blue", `df = 10` = "chartreuse3", `df = 20` = "purple")

ggplot(data = college_data, aes(x = terminal, y = outstate)) +
    geom_point(color = rgb(0.2, 0.4, 0.2, 0.5)) + geom_line(aes(x = terminal,
    y = pred, color = "GCV df"), data = ss_pred_df, size = 1) +
    geom_line(aes(x = terminal, y = pred, color = "df = 2"),
        data = ss_pred_df2, size = 1) + geom_line(aes(x = terminal,
    y = pred, color = "df = 3"), data = ss_pred_df3, size = 1) +
    geom_line(aes(x = terminal, y = pred, color = "df = 6"),
        data = ss_pred_df6, size = 1) + geom_line(aes(x = terminal,
    y = pred, color = "df = 10"), data = ss_pred_df10, size = 1) +
    geom_line(aes(x = terminal, y = pred, color = "df = 20"),
        data = ss_pred_df20, size = 1) + labs(color = "Degrees of Freedom") +
    scale_color_manual(values = colors)
```



From the above model, we can observe that the degrees of freedom from GCV is 4.469, with the smoothing

parameter of 0.844, and λ of 0.039.

I also tried a range of degrees of freedom which was consisted of 2, 3, 6, 10, and 20.

Results show that the higher degree of freedom, the more wiggely the fitted line is. The soomthest line possible is a straight line with the degree of freedom of 2. when we move from df of 2 to df of 3, the line obtains more curvture, but it still doesnt capture a lot of points and we loose a lot of information of non-linearity due to underfitting and increasing the bias. When we choose df of 6, the curve starts to get wiggely, and it intensifies as we increase df to 10 and 20 which is the problem of overfitting and increasing the variance. We can see that the df chosen through GCV (4.469) produces the line that fits the data the best withought having the problem of over or under fitting and the best bias-variance balance. This result visually supports the result that we have obtained from smooth.spline() function. The plot shows that the out of state tuition increases with a very low slope up to the terminal value of 60%. However, after we pass the 60% of faculties with terminal degree, there is and exponential increasing trend in the out of tuition of universities.

Part c

In this part, I would like to fit a **generalized additive model (GAM)** using all the predictors. Refering to previous parts, I have two possible models that I would like to decide between these nested models using ANOVA:

- All variables, with one smooth term: *terminal* (because it was asked about in **part b**)
- All variables, with three smooth terms: *terminal*, *ph_d*, *expend* (variables that seem to follow non-linear trend)

Please note that I used ANOVA to compare these two models despite the fcat that I know it is not the best approach for comparing models and we prefer using CV methods. I chose this approch because it was used for comparing GAM models in both lecture notes from the class and the textbook.

```
gam_fit1 = gam(outstate ~ s(terminal) + apps + expend + accept +
    enroll + top10perc + top25perc + f_undergrad + p_undergrad +
    room_board + books + personal + ph_d + s_f_ratio + perc_alumni +
    grad_rate, data = college_data)

gam_fit2 = gam(outstate ~ s(terminal) + apps + s(expend) + accept +
    enroll + top10perc + top25perc + f_undergrad + p_undergrad +
    room_board + books + personal + s(ph_d) + s_f_ratio + perc_alumni +
    grad_rate, data = college_data)

anova(gam_fit1, gam_fit2, test = "F")
```

```
## Analysis of Deviance Table
##
## Model 1: outstate ~ s(terminal) + apps + expend + accept + enroll + top10perc +
##     top25perc + f_undergrad + p_undergrad + room_board + books +
##     personal + ph_d + s_f_ratio + perc_alumni + grad_rate
## Model 2: outstate ~ s(terminal) + apps + s(expend) + accept + enroll +
##     top10perc + top25perc + f_undergrad + p_undergrad + room_board +
##     books + personal + s(ph_d) + s_f_ratio + perc_alumni + grad_rate
##   Resid. Df Resid. Dev     Df  Deviance     F    Pr(>F)
## 1    542.37 2026858216
## 2    529.01 1565718034 13.357 461140182 11.74 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(gam_fit2)
```

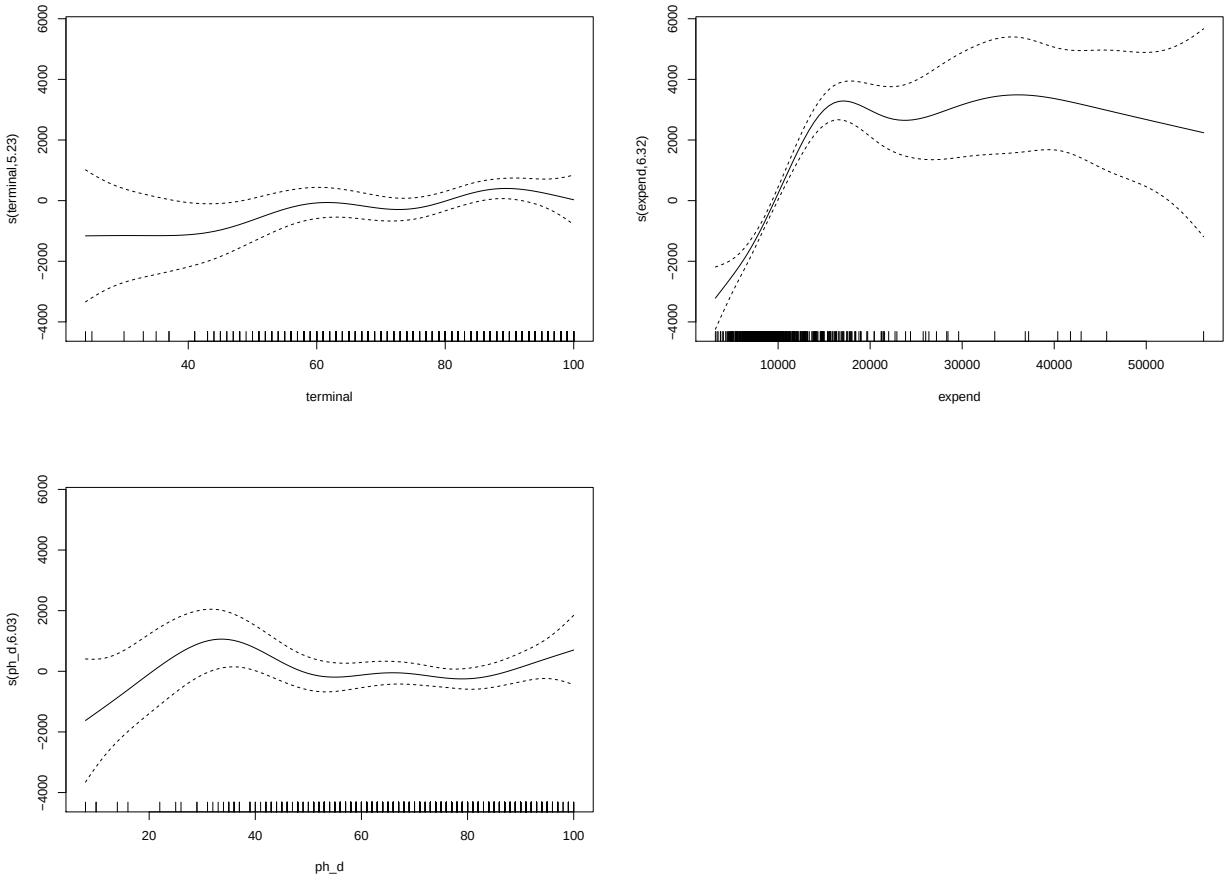
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## outstate ~ s(terminal) + apps + s(expend) + accept + enroll +
##     top10perc + top25perc + f_undergrad + p_undergrad + room_board +
##     books + personal + s(ph_d) + s_f_ratio + perc_alumni + grad_rate
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 5935.99437  732.19340   8.107 3.58e-15 ***
## apps           0.17318    0.10281   1.684  0.09269 .
## accept         0.80471    0.18083   4.450 1.05e-05 ***
## enroll        -3.10672    0.76466  -4.063 5.58e-05 ***
## top10perc     20.48618   13.05764   1.569  0.11726
## top25perc     -3.76999    9.99656  -0.377  0.70623
## f_undergrad    0.01047    0.11988   0.087  0.93045
## p_undergrad   -0.03786    0.12409  -0.305  0.76041
## room_board     0.63302    0.09174   6.900 1.48e-11 ***
## books         -0.29092    0.46385  -0.627  0.53081
## personal      -0.39891    0.12403  -3.216  0.00138 **
## s_f_ratio     46.26842   27.11172   1.707  0.08848 .
## perc_alumni   35.58346    7.64063   4.657 4.05e-06 ***
## grad_rate     24.87097    5.57249   4.463 9.86e-06 ***
## ---
## 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) 5.231  6.352  1.671   0.129
## s(expend)   6.318  7.483 22.851  <2e-16 ***
## s(ph_d)     6.029  7.152  1.914   0.065 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.785   Deviance explained = 79.7%
## GCV = 3.1152e+06  Scale est. = 2.9407e+06  n = 564
```

The results above show that gam_fit2 is better fits compared to gam_fit1 (pvalue <0.05 whoch means we should reject null hypothesis and go with the bigger model). So, the final GAM model that I chose is the following:

$$\begin{aligned} outstae = 5935.99 + 0.80(accept) - 3.11(enroll) + 20.49(top10perc) - 3.76(top25perc) + \\ 0.01(f\_undergrad) - 0.04(p\_undergrad) + 0.63(room\_board) - 0.29(books) - \\ 0.4(personal) + 46.27(sf\_ratio) + 35.58(perc\_alumni) + \\ 24.87(grad\_rate) + s(terminal) + s(expend) + s(phD) \end{aligned}$$

Then I plot the results to see how the smooth terms look like:

```
plot(gam_fit2)
```



We can observe that the degree of freedom obtained from GCV is 5.23 for terminal, 6.32 for expend, and 6.03 for PhD.

Also, all three smooth terms follow non-linear trend, this confirms the non-linear relationship between these three variables and outstate. Also, all the three plots show increasing trends: out of state tuition increases as the percentage of faculty members with terminal degree increase, out of state tuition increases as the instructional expenditure per student increase, and out of state tuition increases as the percentage of faculty with PhD's increase.

Part d

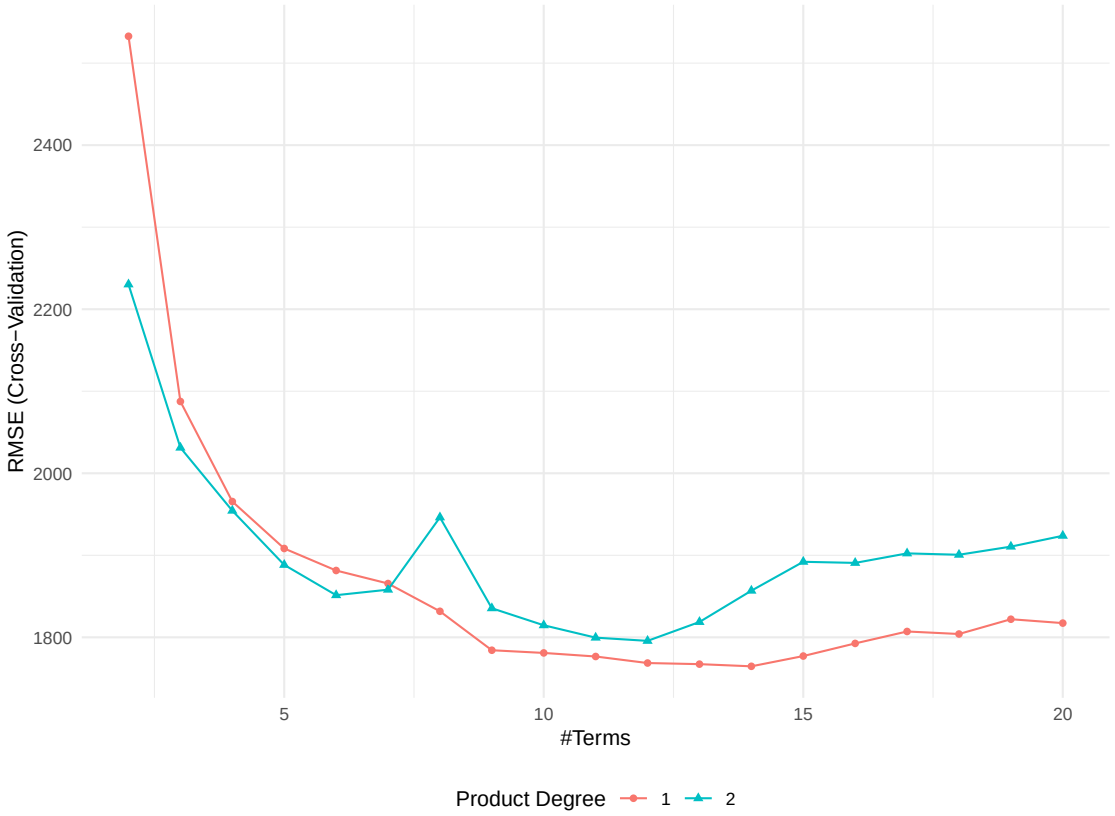
Fit a **Multivariate Adaptive Regression Spline (MARS)** model using all the predictors. Report the final model. Present the partial dependence plot of an arbitrary predictor in your final model.

First, we need to perform a grid search to identify the optimal combination of two tuning parameters (i.e degree of interactions and the number of retained terms) that minimizes the prediction error:

```
mars_grid = expand.grid(degree = 1:2, nprune = 2:20)
set.seed(1)
```

```
mars_fit = train(x, y, method = "earth", tuneGrid = mars_grid,
    trControl = control)

ggplot(mars_fit)
```




```
mars_fit$bestTune
```

```
##    nprune degree
## 13     14      1
```

```
coef(mars_fit$finalModel)  # final model
```

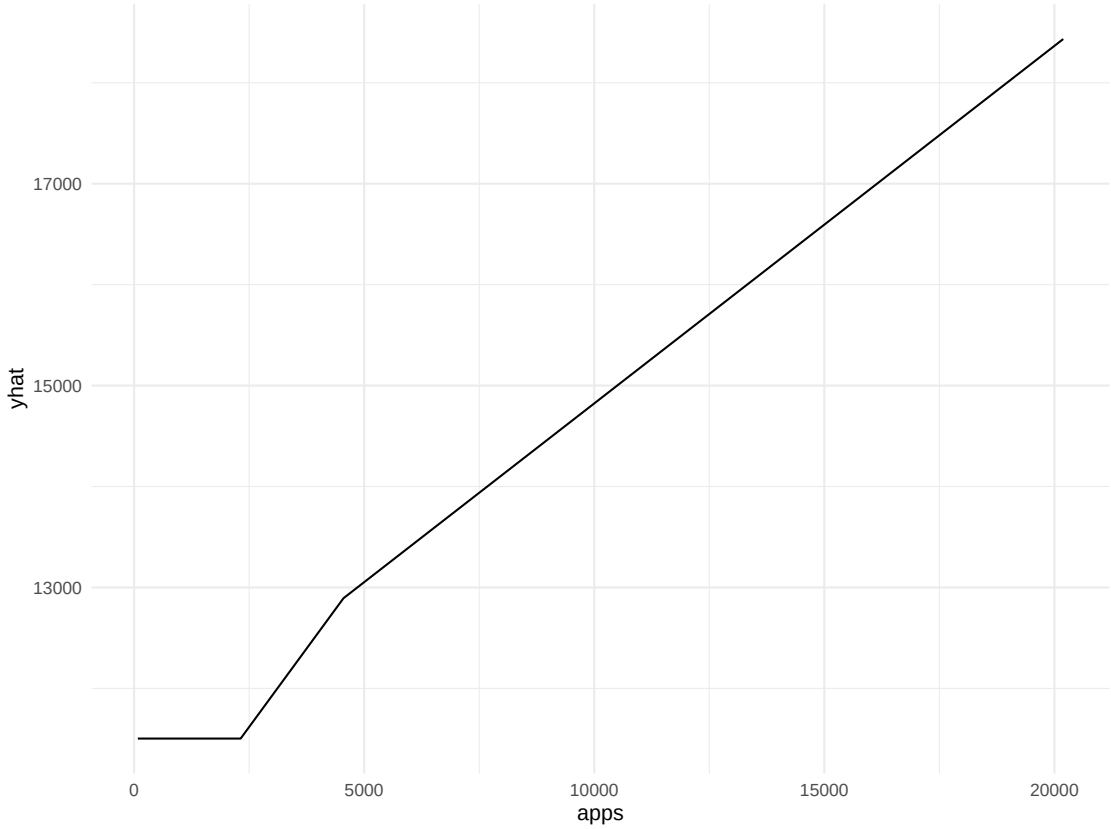
```
##          (Intercept)     h(expend-15365)  h(4450-room_board)     h(grad_rate-97)
##        10661.2171202           -0.7327991           -1.2690141         -240.6920324
##     h(97-grad_rate) h(f_undergrad-1355) h(1355-f_undergrad)   h(22-perc_alumni)
##          -24.5321278           -0.3593198           -1.6238125          -78.4307664
##        h(apps-3712)    h(1300-personal)       h(913-enroll)      h(2193-accept)
##           6.9809000            1.0421976            5.3654412           -1.9460457
##      h(expend-6881)        h(apps-3877)
##           0.7300921           -6.6266893
```

It can be observed that the optimum number of parameters is 14, so our finalModel will have 14 terms including the intercept and 13 hinge functions. Using ESL algorithms that automatically selects the cut points in hinge functions, the final Model looks like this:

$$
\begin{aligned}
outstate = 10661.22 - 1.62h(1355 - f\_undergrad) - 0.36h(f\_undergrad - 1355) + 0.73h(expend - 6881) - \\
0.73h(expend - 15365) - 78.43h(22 - perc\_alumni) - 6.63h(apps - 3877) + 6.98h(apps - 3712) - \\
1.27h(4450 - room\_board) - 240.69h(grad\_rate - 97) + 1.04h(1300 - personal) - \\
24.53h(97 - grad\_rate) + 5.36h(913 - enroll) - 1.95h(2193 - accept)
\end{aligned}
$$

I created partial dependence plot (PDP) for *apps* variable:

```
partial(mars_fit, pred.var = c("apps"), grid.resolution = 10) %>%
    autoplot()
```



Partial dependence plots are used to show the the dependence between the response and a feature, marginalizing over the values of all other features. For example the plot above shows the marginal effect of the number of applications received on the estimated out of state tuition of colleges. However, the plot does not show the value of y for each x value and only gives us the general trend of how the outcome changes with the change of that predictor. For example, we can observe that the overal trend of the estimated outcome is increasing as the number of applications increases.

Part e

In this part, I am going to predict the outstate tuition of Columbia University based on the GAM and MARS models that I have built in **parts c and d**.

```
college_data_cu = read_csv("College.csv") %>% janitor::clean_names() %>%
    filter(college == "Columbia University") %>% select(-college,
    -outstate)
```

```
## Parsed with column specification:
## cols(
##   College = col_character(),
##   Apps = col_double(),
##   Accept = col_double(),
##   Enroll = col_double(),
##   Top10perc = col_double(),
##   Top25perc = col_double(),
##   F.Undergrad = col_double(),
##   P.Undergrad = col_double(),
##   Outstate = col_double(),
##   Room.Board = col_double(),
##   Books = col_double(),
##   Personal = col_double(),
##   PhD = col_double(),
##   Terminal = col_double(),
##   S.F.Ratio = col_double(),
##   perc.alumni = col_double(),
##   Expend = col_double(),
##   Grad.Rate = col_double()
## )
```

```
gam_pred = predict(gam_fit2, newdata = college_data_cu)

mars_pred = predict(mars_fit, newdata = college_data_cu)
```

We can observe that the predicted out of state tuition for Columbia University obtained from GAM model is 18531.36 dollars, and 18455.33 dollars from MARS model. We can see that although the predicted value that is obtained from GAM is 76.03 higher than MARS, these values are in a reasonable range and not very far from each other.