Homework 2

Yiying Wu (yw3996)

R packages

```
library(tidyverse)
library(caret)
library(tidymodels)
library(splines)
```

dataset

```
dat<-read_csv("./data/College.csv")
dat <- na.omit(dat)
```

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

```
set.seed(1)
data_split <- initial_split(dat, prop = 0.80)

# Extract the training and test data
training_data <- training(data_split)
x_train <- training_data %>% select(-College, -Outstate)
y_train <- training_data$Outstate

testing_data <- testing(data_split)
x_test <- testing_data %>% select(-College, -Outstate)
y_test <- testing_data$Outstate

# ctrl
ctrl <- trainControl(method = "cv", number = 10)
```

Outcome variable: Outstate

(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. Describe the observed patterns that emerge with varying degrees of freedom. Select an appropriate degree of freedom for the model and plot this optimal fit. Explain the criteria you used to determine the best choice of degree of freedom.

```
range(dat$perc.alumni)
```

```
## [1]  2 64
```

```
perc.alumni.grid <- seq(from = 0, to = 65, by = 1)

# plot the scatter plot for the training data
p <- ggplot(data = training_data, aes(x = perc.alumni, y = Outstate)) +
     geom_point(color = rgb(.2, .4, .2, .5))
```

For degree of freedom = 3

```
fit.ss3 <- smooth.spline(training_data$perc.alumni, training_data$Outstate,df=3)

pred.ss3 <- predict(fit.ss3,
                   x = perc.alumni.grid)

pred.ss.df3 <- data.frame(pred = pred.ss3$y,
                         perc.alumni = perc.alumni.grid)
```

For degree of freedom = 5

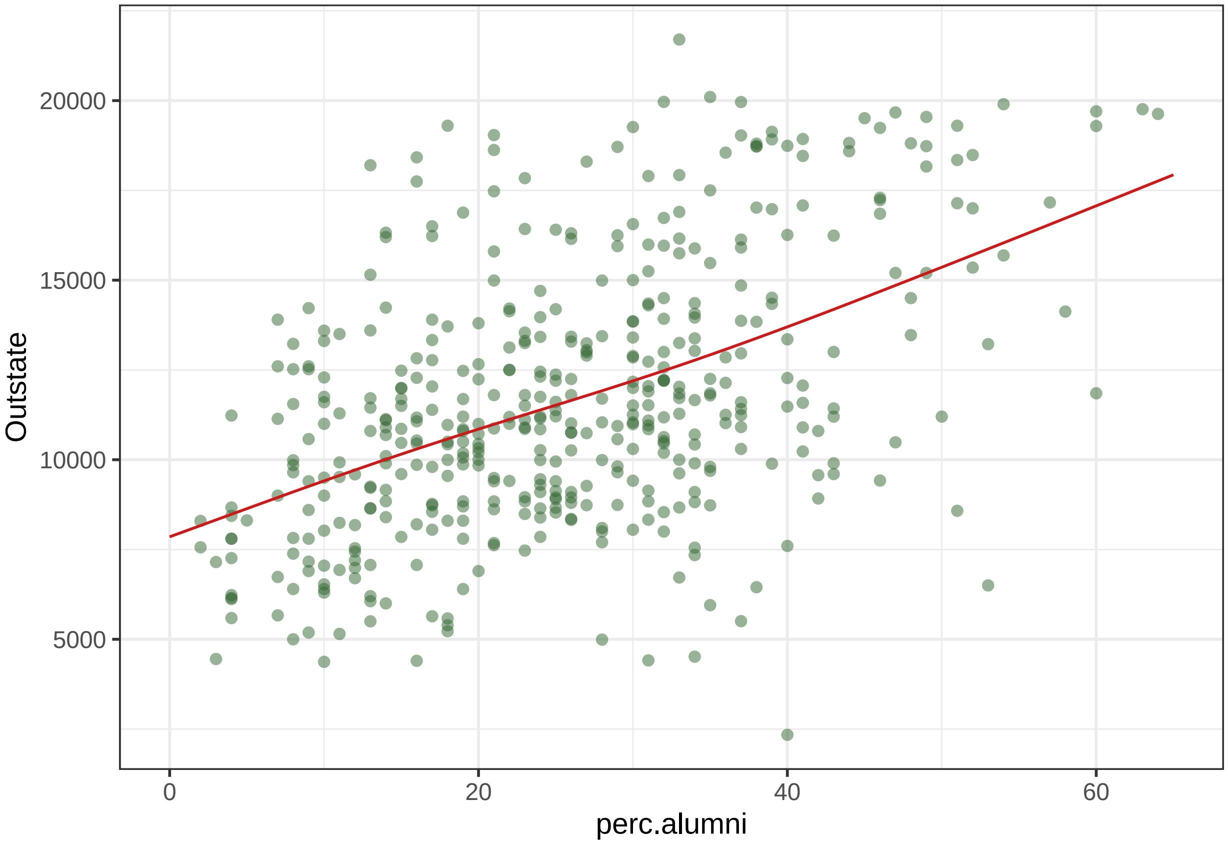
```
fit.ss5 <- smooth.spline(training_data$perc.alumni, training_data$Outstate,df=5)

pred.ss5 <- predict(fit.ss5,
                   x = perc.alumni.grid)

pred.ss.df5 <- data.frame(pred = pred.ss5$y,
                         perc.alumni = perc.alumni.grid)
```

For degree of freedom = 8

```
fit.ss8 <- smooth.spline(training_data$perc.alumni, training_data$Outstate,df=8)

pred.ss8 <- predict(fit.ss8,
                   x = perc.alumni.grid)

pred.ss.df8 <- data.frame(pred = pred.ss8$y,
                         perc.alumni = perc.alumni.grid)
```

Plot the model fits for df=3

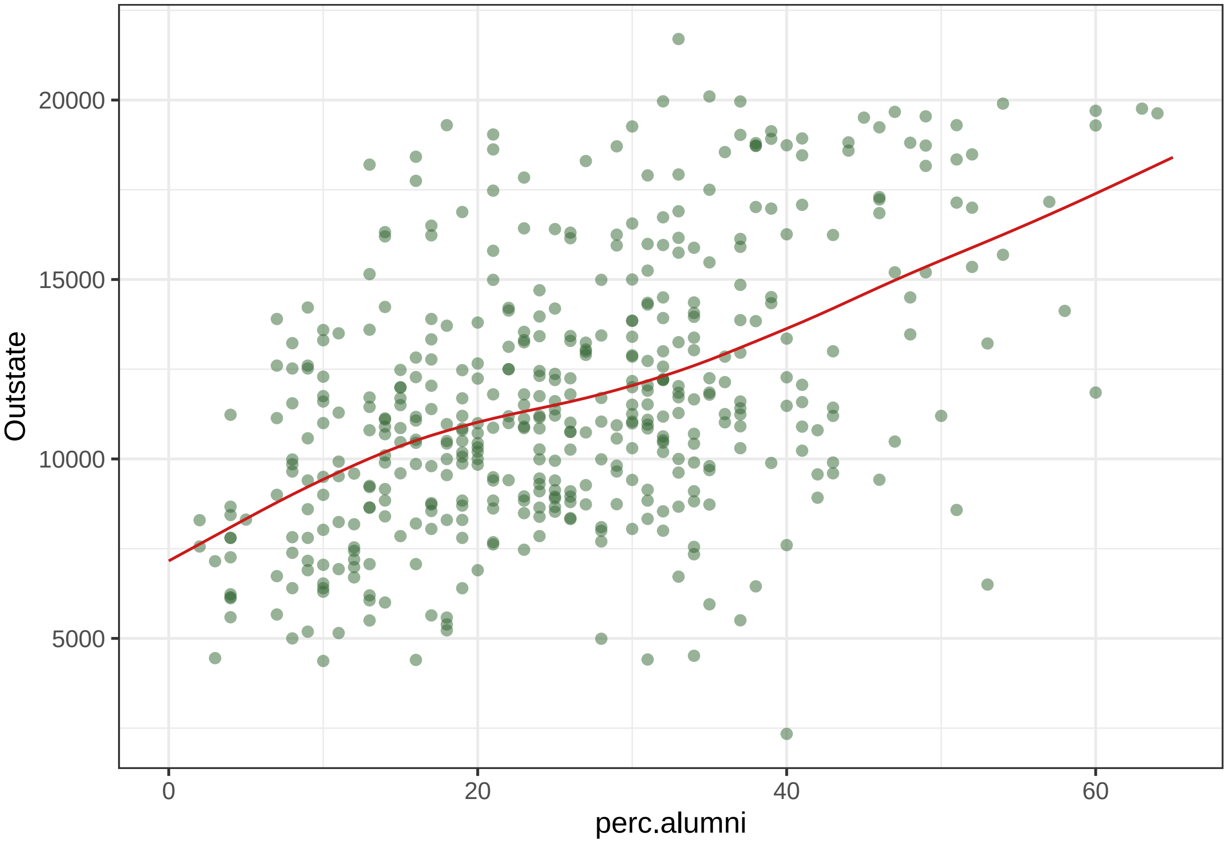
```
p +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df3,
          color = rgb(.8, .1, .1, 1)) + theme_bw()
```



Description From the plot, there seems to be a positive correlation between the percentage of alumni donors and out-of-state tuition costs.The relationship seems almost linear.

Plot the model fits for df=5

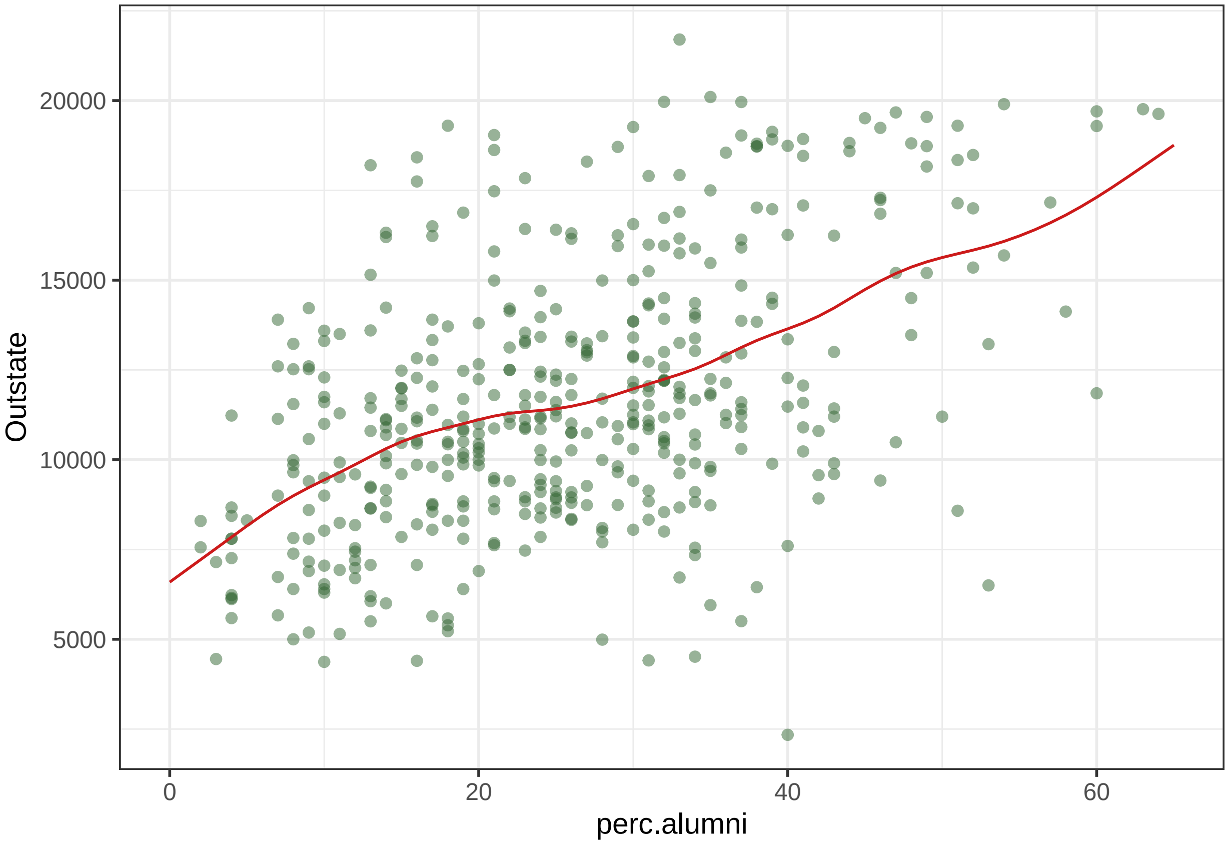
```
p +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df5,
          color = rgb(.8, .1, .1, 1)) + theme_bw()
```



Description In this plot, the fitted line seems to curve slightly upward from 5 to 30, suggesting a possible non-linear relationship. This might imply that the rate at which out-of-state tuition increases becomes greater at higher percentages of alumni donors within this range.

Plot the model fits for df=8

```
p +
geom_line(aes(x = perc.alumni, y = pred), data = pred.ss.df8,
          color = rgb(.8, .1, .1, 1)) + theme_bw()
```



Description This plot indicates a nonlinear pattern.

Select an appropriate degree of freedom for the model

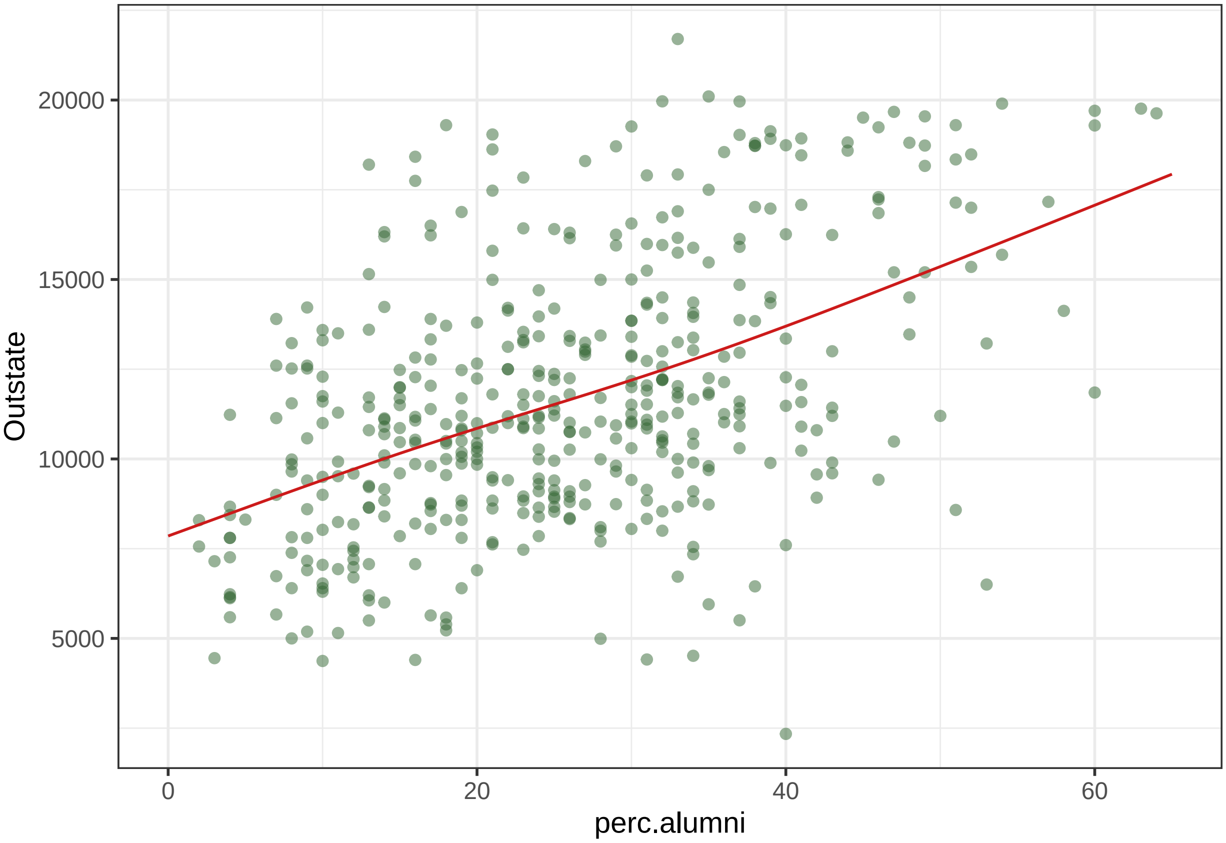
```
# does generalized-cv to determine df automatically
fit.ss <- smooth.spline(training_data$perc.alumni, training_data$Outstate,df=3)

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

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

# print the selected df
fit.ss$df
```

```
## [1] 3.000318
```

Therefore, the appropriate degree of freedom for the model is 3.0003

plot this optimal fit

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

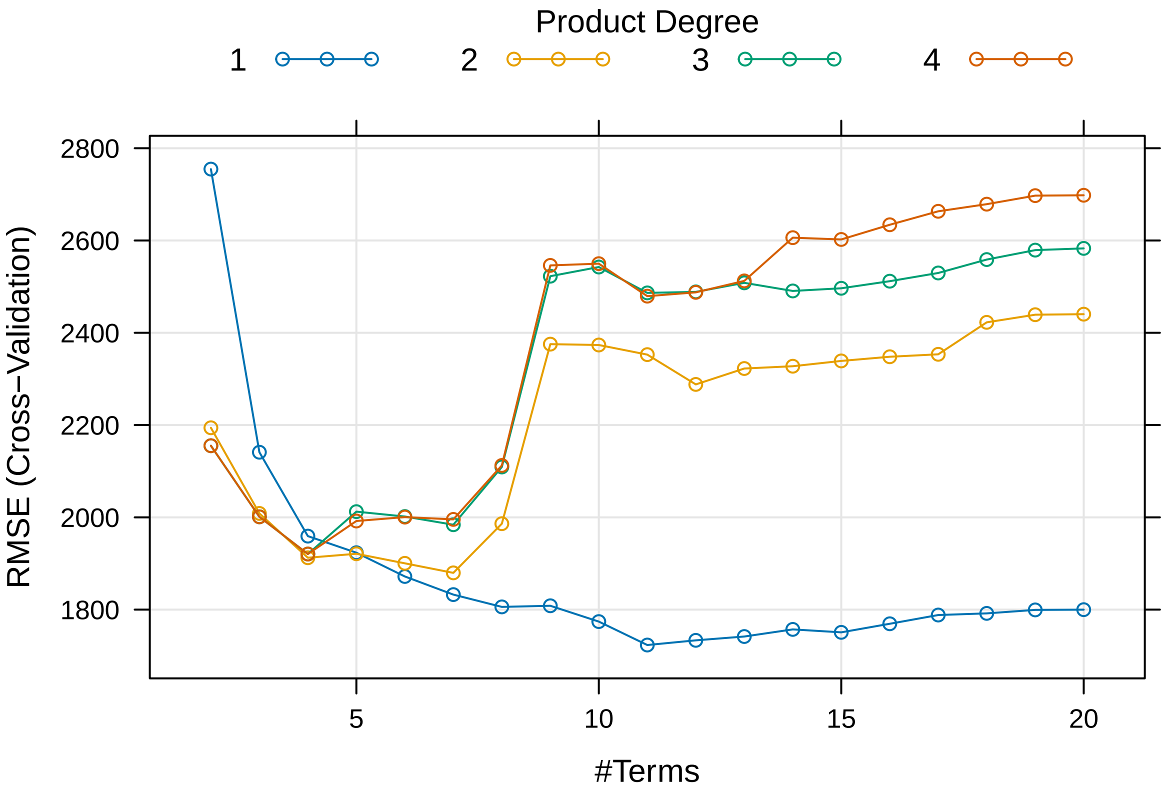


The criteria used to determine the best choice of degree of freedom is **generalized cross validation**.

(b)

Train a multivariate adaptive regression spline (MARS) model to predict the response variable. Report the regression function. Present the partial dependence plot of an arbitrary predictor in your model. Report the test error.

```
set.seed(1)
model.mars <- train(x = x_train,
                    y = y_train,
                    method = "earth", # earth is for mars
                    tuneGrid = expand.grid(degree = 1:4,
                                           nprune = 2:20),
                    trControl = ctrl)
# degree from 1~4 is sufficient
# nprune can be larger than the number of predictors, make it as large as possible

plot(model.mars)
```



```
# both number of terms and product degree are upper bounds

# best tune
model.mars$bestTune
```

```
##    nprune degree
## 10     11      1
```

```
coef(model.mars$finalModel)
```

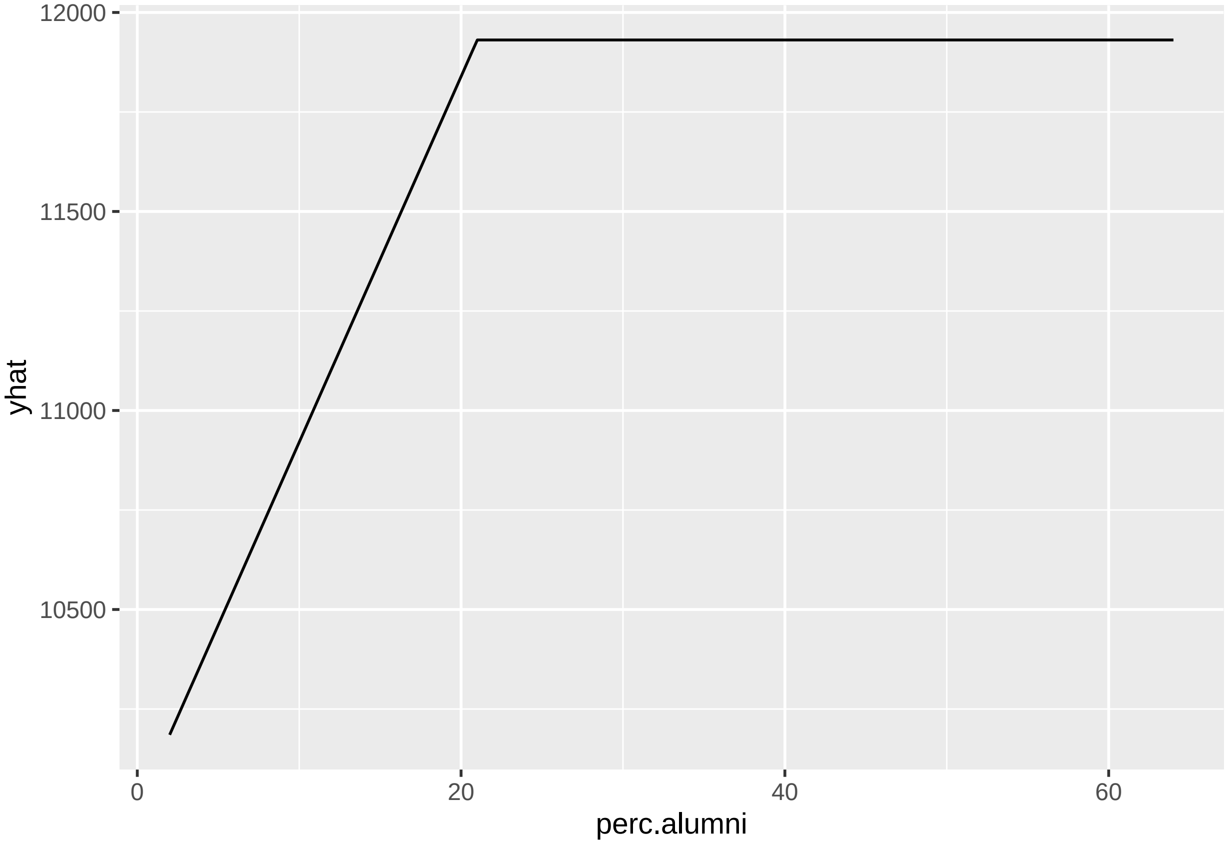
```
##         (Intercept)     h(Expend-15494)     h(81-Grad.Rate)  h(4138-Room.Board)
##       10252.3200044          -0.7594586         -29.5255903          -1.5537526
## h(F.Undergrad-1251) h(1251-F.Undergrad)    h(21-perc.alumni)        h(Apps-2694)
##          -0.3243249          -1.5277676         -91.8812340           0.3838076
##       h(942-Enroll)      h(Expend-6898)       h(2081-Accept)
##           5.1925945           0.7691120          -2.0869169
```

The **regression function** is

$$
\begin{aligned}
f(x) =& 10252.32 - 0.76 \cdot h(Expend - 15494) - 29.53 \cdot h(81 - Grad.Rate) - 1.55 \cdot h(4138 - Room.Board) \\
& - 0.32 \cdot h(F.Undergrad - 1251) - 1.53 \cdot h(1251 - F.Undergrad) - 91.88 \cdot h(21 - perc.alumni) \\
& + 0.38 \cdot h(Apps - 2694) + 5.19 \cdot h(942 - Enroll) + 0.77 \cdot h(Expend - 6898) - 2.09 \cdot h(2081 - Accept)
\end{aligned}
$$

partial dependence plot of an arbitrary predictor (perc.alumni) in the model.

```
pdp::partial(model.mars, pred.var = c("perc.alumni"), grid.resolution = 200) %>% autoplot()
```



test error

```
mars.pred <- predict(model.mars, newdata = x_test)
test_error_mars <- mean((mars.pred - y_test)^2)
test_error_mars
```

```
## [1] 2990281
```

```
RMSE_mars <- sqrt(test_error_mars)
RMSE_mars
```

```
## [1] 1729.243
```

The RMSE of MARS model is 1729.24