

P8106_HW2

Naomi Simon-Kumar

ns3782

02/03/2025

Contents

Loading libraries	1
Partition into training and testing set	1
Question (a): Smoothing spline models	2
Question (b): Multivariate Adaptive Regression Spline (MARS) model	5

Loading libraries

```
library(tidyverse)
library(ISLR)
library(pls)
library(caret)
library(tidymodels)
```

Partition into training and testing set

```
# Read in dataset
college <- read.csv("College.csv")

# Remove NAs
college <- na.omit(college)

# Set seed for reproducibility
set.seed(299)

# Split data into training and testing data
data_split <- initial_split(college, prop = 0.8)

# Extract the training and test data
training_data <- training(data_split)
testing_data <- testing(data_split)
```

Question (a): Smoothing spline models

```
# Set seed for reproducibility
set.seed(299)

# Fit smoothing spline
fit.ss <- smooth.spline(training_data$perc.alumni, training_data$Outstate)

# Define a grid for smooth predictions using dataset range
# Will illustrate curves beyond dataset boundary
perc.alumni.grid <- seq(from = min(training_data$perc.alumni) - 10,
                        to = max(training_data$perc.alumni) + 10,
                        by = 1)

# Create dataframe of degrees of freedom range
df_values <- seq(3, 15, by = 1)

# Create dataframe to store smoothing spline predictions
ss.predictions <- data.frame()

# Use for loop to populate ss.predictions dataframe
# Code Source: https://www.rdocumentation.org/packages/openintro/versions/2.4.0/topics/loop

for (df in df_values) {

  # Plot smoothing spline curves for different degrees of freedom
  # Code Source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/smooth.spline

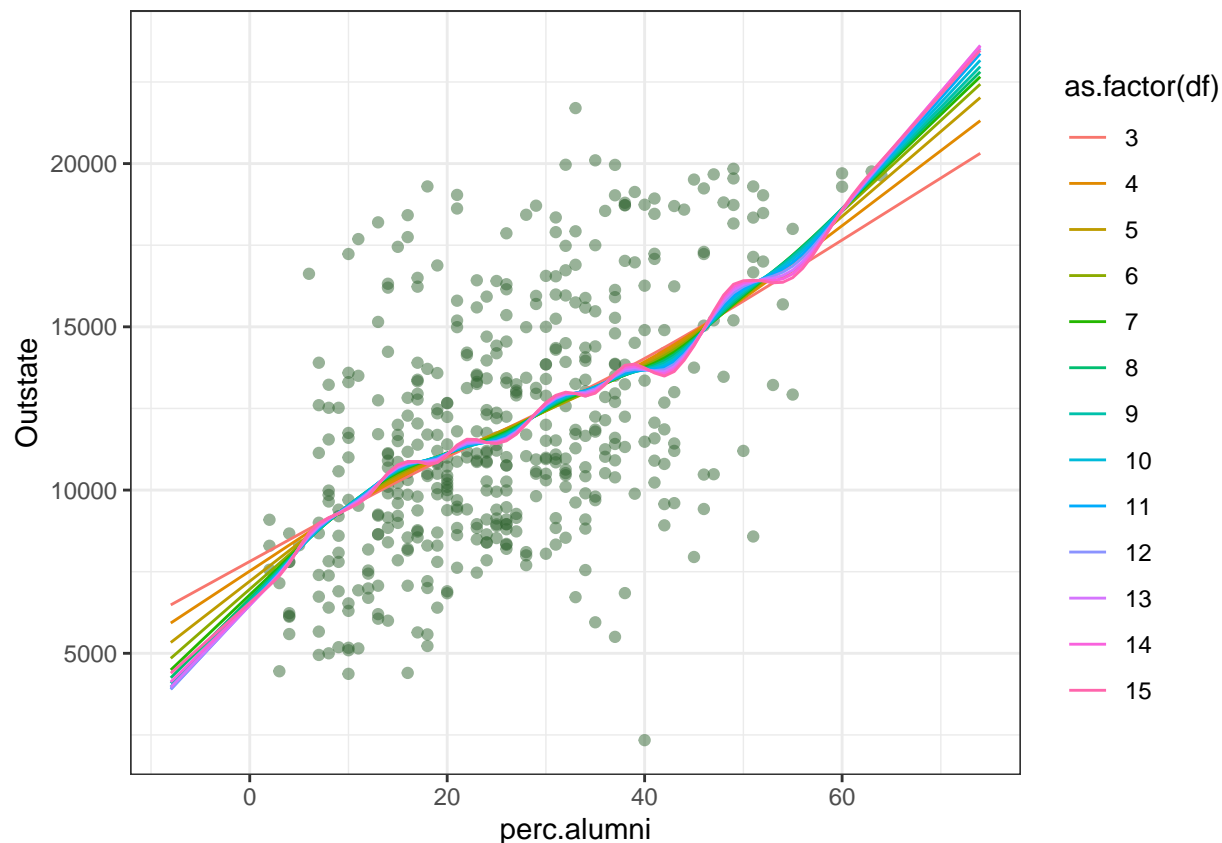
  fit <- smooth.spline(training_data$perc.alumni, training_data$Outstate, df = df)
  pred <- predict(fit, x = perc.alumni.grid)

  # Store pred for current df
  temp_df <- data.frame(
    perc.alumni = perc.alumni.grid,
    pred = pred$,
    df = df
  )

  ss.predictions <- rbind(ss.predictions, temp_df) # Add to ss.predictions
}

# Scatter plot of perc.alumni vs Outstate
ss.p <- ggplot(training_data, aes(x = perc.alumni, y = Outstate)) +
  geom_point(color = rgb(.2, .4, .2, .5))

# Add smoothing spline curves for df range 3-15
ss.p +
  geom_line(aes(x = perc.alumni, y = pred, color = as.factor(df)),
            data = ss.predictions) + theme_bw()
```



The plot I produced represents the fitted smoothing spline curves for each degree of freedom between the range of 3 and 15. It can be observed that as the degrees of freedom increases over this range, the smoothing spline goes from underfitting, to overfitting of the data. Specifically, as the degrees of freedom increases beyond ~10, the spline curve becomes highly wiggly, particularly at extreme values of perc.alumni, indicating overfitting of the data. For lower df (i.e., 3–5), the spline appears quite smooth.

```
# Set seed for reproducibility
set.seed(299)

# Fits a smoothing spline with automatically selected df using generalized cross-validation
fit.ss.gcv <- smooth.spline(training_data$perc.alumni, training_data$Outstate)

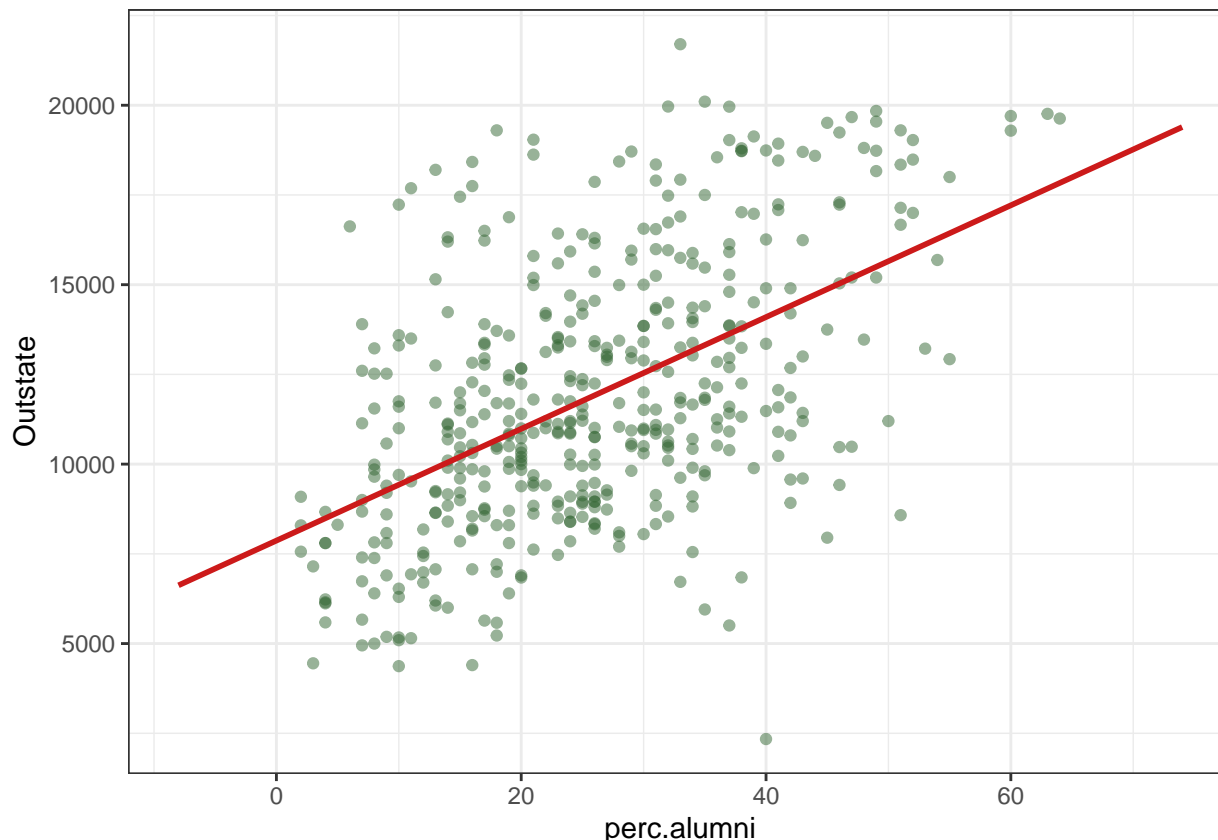
gcv_df <- fit.ss.gcv$df # Extract optimal df selected by GCV
print(gcv_df) # 2.000237 appears to be the optimal df selected by GCV
```

```
## [1] 2.000237
```

```
# Predict values using GCV-selected df
pred.ss.gcv <- predict(fit.ss.gcv, x = perc.alumni.grid)

# Convert predictions to dataframe
ss.optimal <- data.frame(
  perc.alumni = perc.alumni.grid,
  pred = pred.ss.gcv$y
)
```

```
# Add optimal spline curve to the plot
ss.p +
  geom_line(aes(x = perc.alumni, y = pred), data = ss.optimal,
    color = rgb(.8, .1, .1, 1), linewidth = 1) + theme_bw()
```



```
# Checking lambda value

lambda_value <- fit.ss.gcv$lambda
print(lambda_value) # High lambda value 2384.249
```

```
## [1] 2384.249
```

For my smoothing spline model, I selected a degree of freedom of approximately 2, as determined by Generalized Cross-Validation (GCV). The GCV method minimises a criterion balancing model fit against model complexity, and in this case, selected a nearly linear model (df approx. 2.000237). This is evident in the plot above where the optimal smoothing spline appears as essentially a straight line, indicating that the relationship between percentage of alumni who donate and out-of-state tuition is best represented linearly.

While I explored models with higher degrees of freedom (ranging from 3 to 15), these more complex models with greater flexibility showed increasing wiggleness, especially at extreme values of perc.alumni as discussed, indicating potential overfitting. The GCV criterion effectively penalised this unnecessary complexity and favored the simpler model. From what we have covered in class, we know that smoothing splines can range from very flexible (high df) to very smooth, with a minimum of 2 degrees of freedom. As discussed in class, when lambda approaches infinity, the function becomes linear, and this model has a relatively large lambda value (2384.249), explaining why the optimal model with df approximately 2 appears as a straight line. This

relationship between lambda and degrees of freedom demonstrates how the roughness penalty controls the smoothness of the function.

The GCV has selected a model at approximately this minimum value, resulting in a nearly linear fit that would indicate out-of-state tuition (outstate) increases steadily with the percentage of alumni who donate (perc.alumni), without needing to be represented by a more complex nonlinear relationship.

Question (b): Multivariate Adaptive Regression Spline (MARS) model