Nonlinear Models and Tree-based Methods

Justin Lo

Non-linear models

This question relates to the `College` dataset from the `ISLR` package. Start by loading that package, as well as the `gam` package which will allow us to estimate generalised additive models, the `splines` package (which, unsurprisingly, let's us estimate splines), and the `randomForest` package (guess what that is for?).

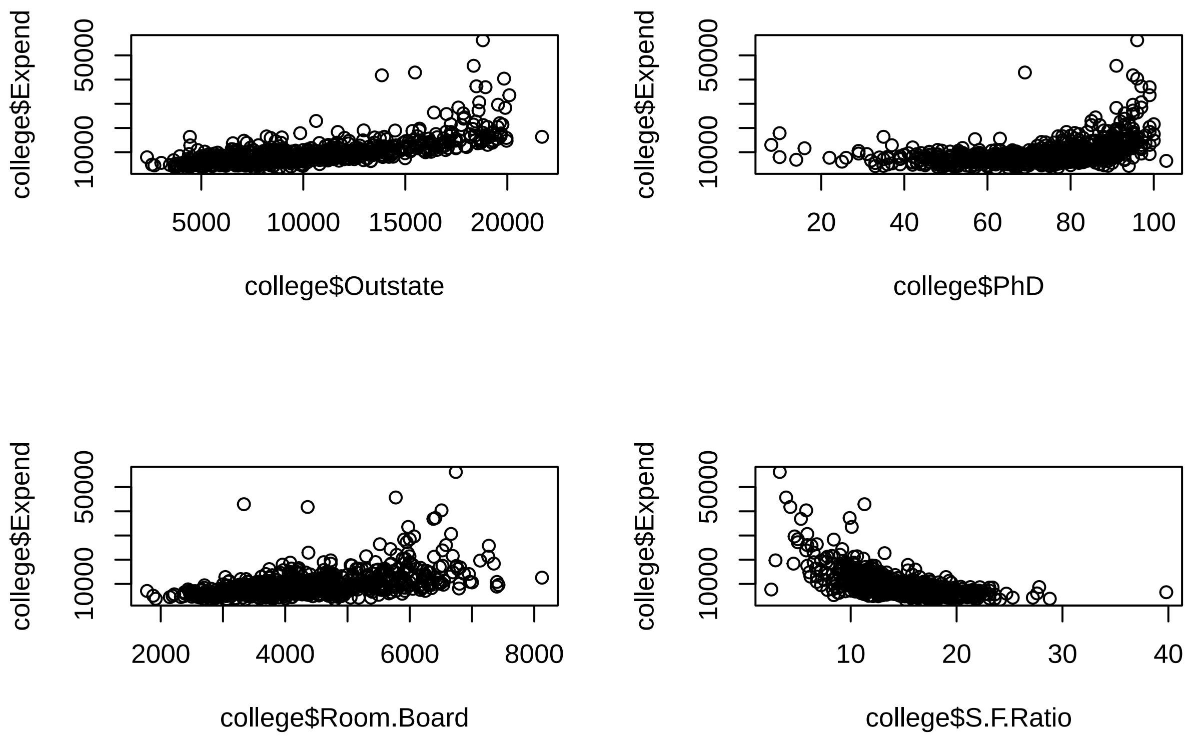
```
library(ISLR)
library(gam)
library(splines)
library(randomForest)
library(tidyverse)
```

The `College` data contains several variables for 777 US Colleges in the 1990s. Look at the help file for this data set (`?College`) for a description of the variables that are included.

In this seminar, we will be experimenting with different approaches to estimating non-linear relationships between the `Expend` variable – which measures the expenditure per student of each college (in dollars) – and several other variables in the data.

```
college<-College
par(mfrow=c(2,2))
plot(college$Outstate, college$Expend)
plot(college$PhD, college$Expend)
plot(college$Room.Board, college$Expend)
plot(college$S.F.Ratio, college$Expend)
```

a. Create a series of scatter plots which show the association between the `Expend` variable and the following four predictors: `Outstate`, `PhD`, `Room.Board`, and `S.F.Ratio`. For which of these variables do you think there is evidence of a non-linear relationship?




```r
regression_model_1<- college %>%
  lm(Expend ~ poly(Outstate, 2), data = .)
summary(regression_model_1)
```

b. Estimate four regression models, all with `Expend` as the outcome variable, and each including one of the predictors you plotted above. Include a second-degree polynomial transformation of X in each of the models (you can do this by using `poly(x_variable,2)` in the `lm()` model formula). Interpret the significance of the squared term in each of your models. Can you reject the null hypothesis of linearity?

```
##
## Call:
## lm(formula = Expend ~ poly(Outstate, 2), data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9328  -1636   -497    665  36680
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9660.2      130.5   74.04   <2e-16 ***
## poly(Outstate, 2)1   97863.5     3636.7   26.91   <2e-16 ***
## poly(Outstate, 2)2   36675.2     3636.7   10.09   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3637 on 774 degrees of freedom
## Multiple R-squared:  0.5162, Adjusted R-squared:  0.515
## F-statistic: 412.9 on 2 and 774 DF,  p-value: < 2.2e-16
```

```
regression_model_2<- college%>%
  lm(Expend ~ poly(PhD, 2), data = .)
summary(regression_model_2)
```

```
##
## Call:
## lm(formula = Expend ~ poly(PhD, 2), data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -12750  -2263   -357   1309  40415
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      9660.2      154.4   62.55   <2e-16 ***
## poly(PhD, 2)1  62950.2     4304.9   14.62   <2e-16 ***
## poly(PhD, 2)2  53405.8     4304.9   12.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4305 on 774 degrees of freedom
## Multiple R-squared:  0.3221, Adjusted R-squared:  0.3203
## F-statistic: 183.9 on 2 and 774 DF,  p-value: < 2.2e-16
```

```
regression_model_3<- college%>%
  lm(Expend ~ poly(Room.Board, 2), data = .)
summary(regression_model_3)
```

```
##
## Call:
## lm(formula = Expend ~ poly(Room.Board, 2), data = .)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9804  -2240   -628   1293  40005
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             9660.2      161.6  59.784   <2e-16 ***
## poly(Room.Board, 2)1  72983.8     4504.1  16.204   <2e-16 ***
## poly(Room.Board, 2)2  11416.1     4504.1   2.535   0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4504 on 774 degrees of freedom
## Multiple R-squared:  0.2579, Adjusted R-squared:  0.256
## F-statistic: 134.5 on 2 and 774 DF,  p-value: < 2.2e-16
```

```r
regression_model_4<- college%>%
  lm(Expend ~ poly(S.F.Ratio, 2), data = .)
summary(regression_model_4)
```

```
##
## Call:
## lm(formula = Expend ~ poly(S.F.Ratio, 2), data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19209.3  -1896.6   -415.4   1556.7  31145.9
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           9660.2      138.5   69.76   <2e-16 ***
## poly(S.F.Ratio, 2)1 -84925.2     3860.2  -22.00   <2e-16 ***
## poly(S.F.Ratio, 2)2  49124.6     3860.2   12.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3860 on 774 degrees of freedom
## Multiple R-squared:  0.4549, Adjusted R-squared:  0.4535
## F-statistic:   323 on 2 and 774 DF,  p-value: < 2.2e-16
```

```r
attach(college)
range(PhD)
fitted_vals_quadratic <- predict(regression_model_2, newdata = data.frame(PhD= 8:103))
#It predicts the values for a new data frame with the PhD variable ranging from 8 to 103.
plot(PhD, Expend,
     xlab = "Percentage of faculty with a PhD",
     ylab = "Predicted Expenditure per Student")
lines(8:103, fitted_vals_quadratic, col = "red")
#This line adds a line plot to the existing scatter plot. The lines() function is used to draw a line c
```

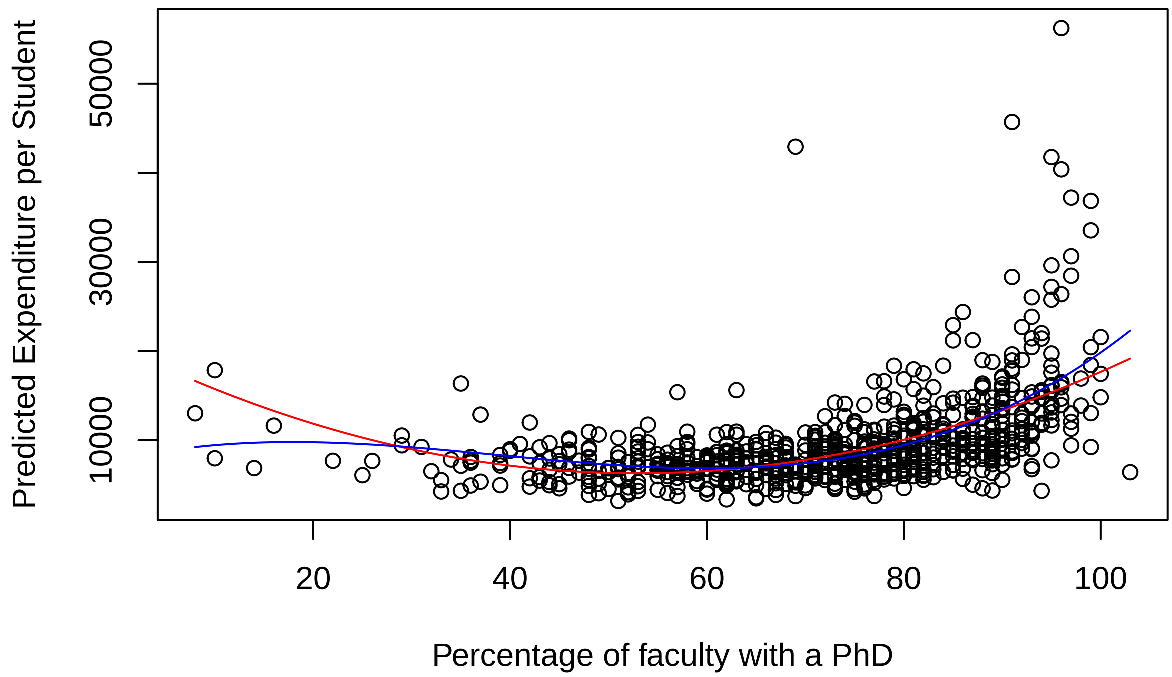
c. Using the regression you estimated in part b to describe the association between `Expend` and `PhD`, calculate fitted values across the range of the `PhD` variable. Recreate the plot between these variables that you constructed in part a, and add a line representing these fitted values to the plot (using the `lines()` function) to illustrate the estimated relationship. Interpret the graph. (You will need to use the `predict()` function with the `newdata` argument in order to complete this question. You can also find the range of the `PhD` variable using the `range()` function.) I have given you some starter code below.

```r
regression_model_5<- lm(Expend ~ poly(PhD,3), data=college)
summary(regression_model_5)
```

d. Re-estimate the `Expend ~ PhD` model, this time including a cubic polynomial (i.e. of degree 3). Can you reject the null hypothesis for the cubic term? Add another line (in a different colour) with the fitted values from this model to the plot you created in part c.

```
##
## Call:
## lm(formula = Expend ~ poly(PhD, 3), data = college)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -15884  -2266   -373   1330  39272
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)       9660        152  63.544  < 2e-16 ***
## poly(PhD, 3)1    62950       4238  14.855  < 2e-16 ***
## poly(PhD, 3)2    53406       4238  12.603  < 2e-16 ***
## poly(PhD, 3)3    21518       4238   5.078 4.79e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4238 on 773 degrees of freedom
## Multiple R-squared:  0.344,  Adjusted R-squared:  0.3414
## F-statistic: 135.1 on 3 and 773 DF,  p-value: < 2.2e-16
```

```r
fitted_vals_cubic<- predict(regression_model_5, newdata = data.frame(PhD=8:103))

fitted_vals_quadratic <- predict(regression_model_2, newdata = data.frame(PhD= 8:103))
attach(college)
plot(PhD, Expend,
     xlab = "Percentage of faculty with a PhD",
     ylab = "Predicted Expenditure per Student")
lines(8:103, fitted_vals_quadratic, col = "red")
lines(8:103, fitted_vals_cubic, col = 'blue')
```



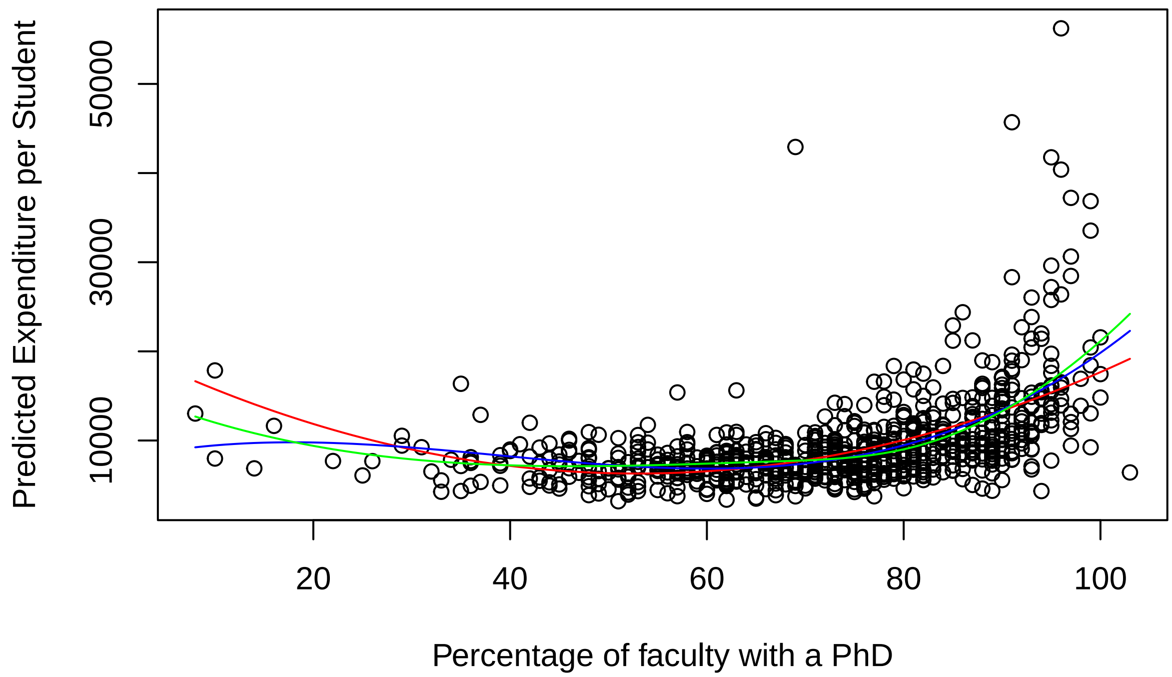
It is statistically significant to reject the H0 for the cubic term.

```
attach(college)
```

e. Estimate a new model for the relationship between `Expend` and `PhD`, this time using a cubic spline instead of a polynomial. You can implement the cubic spline by using the `bs()` function, which is specified thus: `lm(outcome ~ bs(x_variable, df = ?, degree = 3))`. Select a value for the `df` argument that you think is reasonable. Estimate the model and then, again, plot the fitted values across the range of the `PhD` variable.

```
## The following objects are masked from college (pos = 3):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc
```

```
regression_model_6<- lm(Expend ~ bs(PhD, df = 5, degree = 3))
fitted_vals_cubic_spline<- predict(regression_model_6, newdata=data.frame(PhD=8:103))
plot(PhD, Expend,
     xlab = "Percentage of faculty with a PhD",
     ylab = "Predicted Expenditure per Student")
lines(8:103, fitted_vals_quadratic, col = "red")
lines(8:103, fitted_vals_cubic, col = 'blue')
lines(8:103, fitted_vals_cubic_spline, col='green')
```



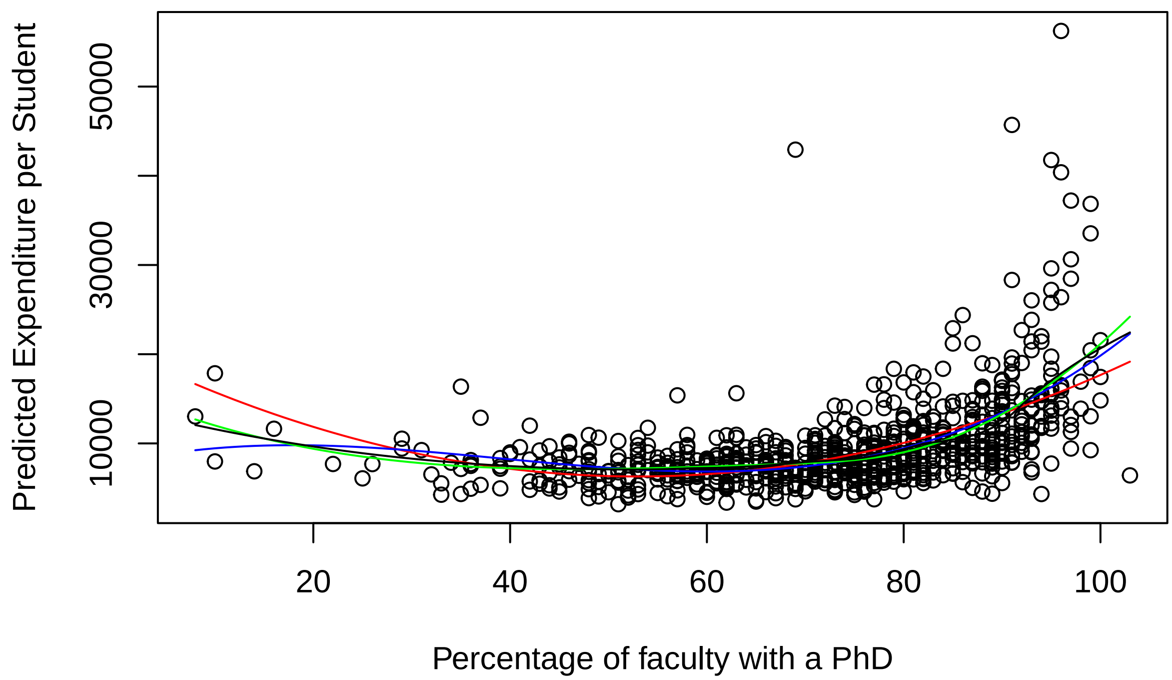

```r
attach(college)
```

f. Guess what? Now it's time to do the same thing again, but this time using a `loess()` model. The key parameter here is the span. High values for span will result in a less flexible model, and low values for span will result in a more flexible model. Pick a value that you feel is appropriately wiggly. Again, add it to your (now very colourful) plot.

```
## The following objects are masked from college (pos = 3):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc

## The following objects are masked from college (pos = 4):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc
```

```r
regression_model_7<- loess(Expend ~ PhD, data=college, span=0.4)
fitted_vals_loess<- predict(regression_model_7, newdata = data.frame(PhD= 8:103))
```

```r
plot(PhD, Expend,
     xlab = "Percentage of faculty with a PhD",
     ylab = "Predicted Expenditure per Student")
lines(8:103, fitted_vals_quadratic, col = "red")
lines(8:103, fitted_vals_cubic, col = 'blue')
lines(8:103, fitted_vals_cubic_spline, col='green')
lines(8:103, fitted_vals_loess, col= 'black' )
```



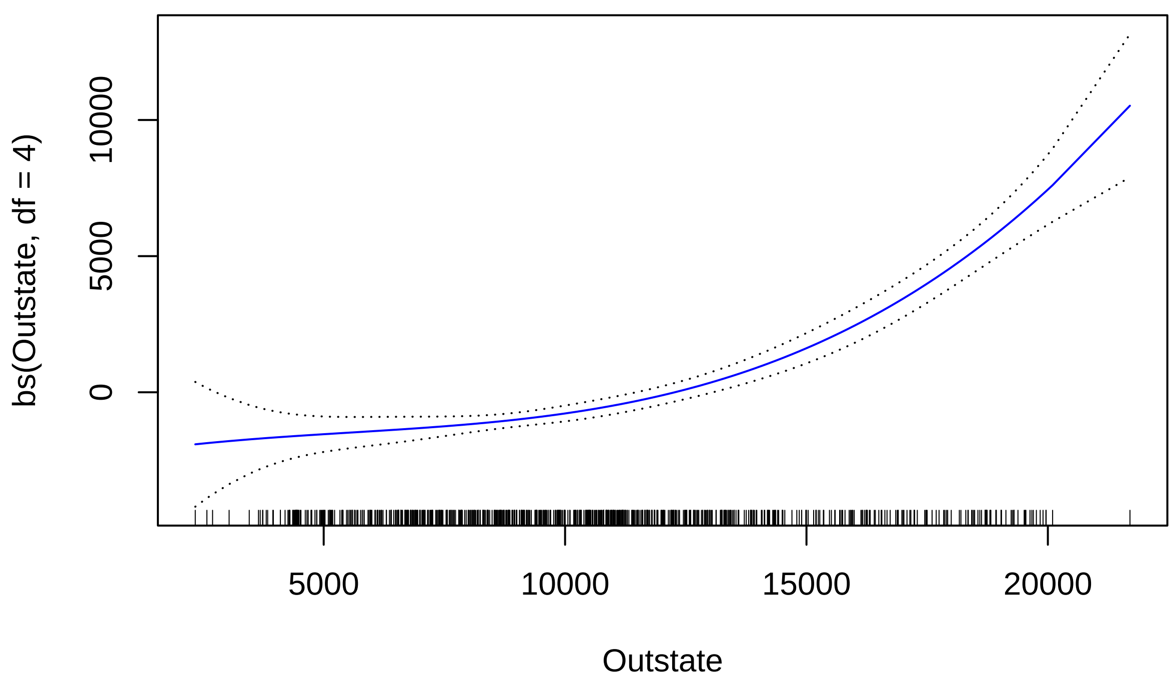
g. Examine the nice plot you have constructed. Which of the lines of fitted values best characterise the relationship between `PhD` and `Expend` in the data? Can you tell? cannot tell purely based on observing plots.

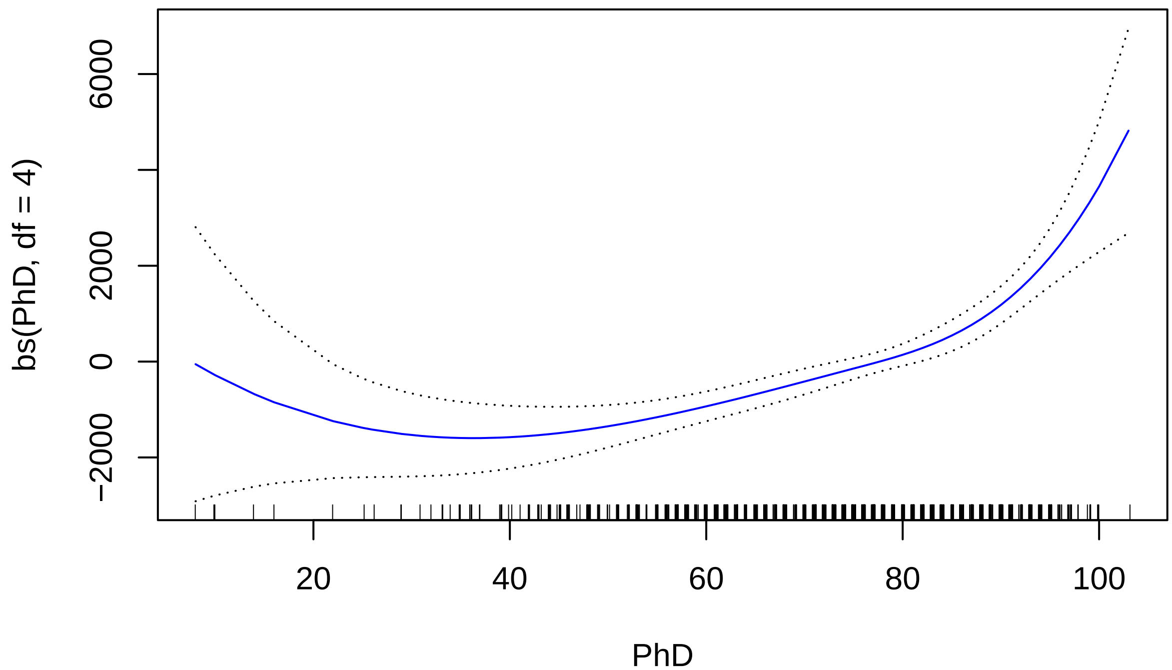
```r
attach(college)
```

h. Fit a generalised additive model (GAM) to the `College` data using the `gam()` function. This model is just like the `lm()` function, but it allows you to include flexible transformations of the covariates in the model. In this example, estimate a GAM with `Expend` as the outcome, and use all four of predictors that you plotted in part a as well as the `Private` variable. For each of the continuous predictors, use a cubic spline (`bs()`) with 4 degrees of freedom. Once you have estimated your model, plot the results by passing the estimated model object to the `plot()` function. Interpret the results.

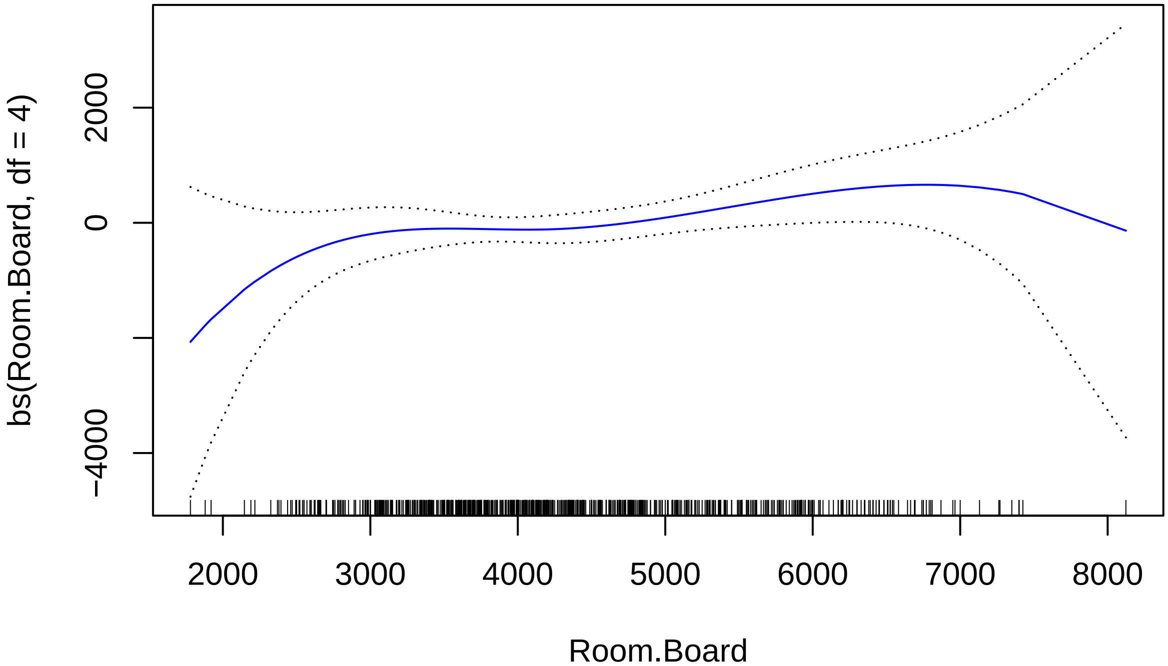
```
## The following objects are masked from college (pos = 3):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc

## The following objects are masked from college (pos = 4):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc

## The following objects are masked from college (pos = 5):
##
##     Accept, Apps, Books, Enroll, Expend, F.Undergrad, Grad.Rate,
##     Outstate, P.Undergrad, perc.alumni, Personal, PhD, Private,
##     Room.Board, S.F.Ratio, Terminal, Top10perc, Top25perc
```

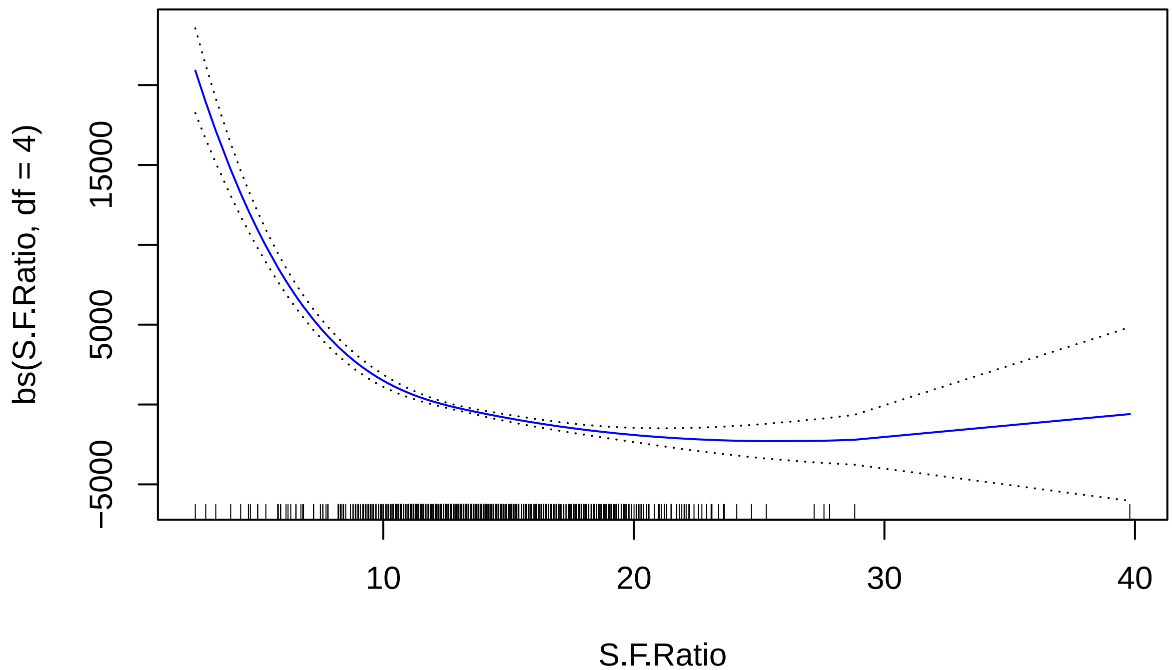
```r
generalised_additive_model<- gam(Expend ~ bs(Outstate, df=4)+ bs(PhD, df=4)+ bs(Room.Board, df=4)+ bs(S

plot(generalised_additive_model, se=TRUE, col="blue")
```

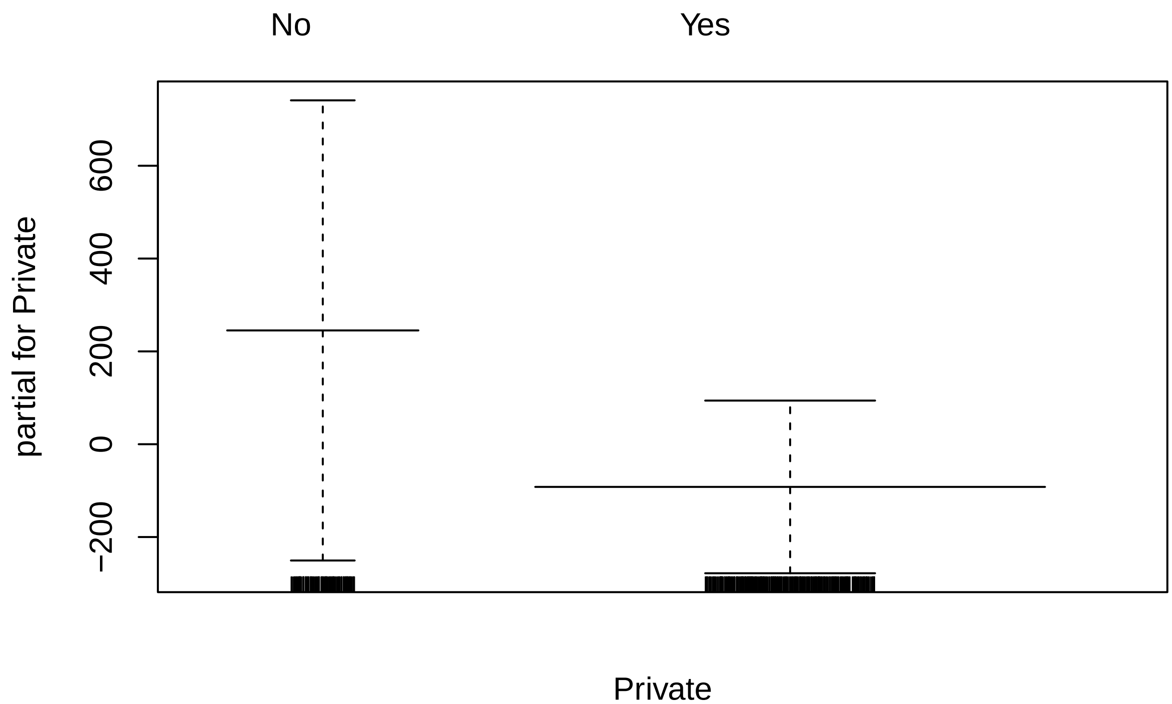




bs(PhD, df = 4)
6000
2000
0
−2000
20
40
60
80
100
PhD


bs(Room.Board, df = 4)
2000
0
−4000
2000
3000
4000
5000
6000
7000
8000
Room.Board


bs(S.F.Ratio, df = 4)
−5000
5000
15000
10
20
30
40
S.F.Ratio




```
#se=true means that standard errors will be displayed on the plots of the fitted smooth terms.
```

Tree-based methods

Apply a) bagging and b) Random Forests to the `Weekly` data set. This dataset includes Weekly percentage returns for the S&P 500 stock index between 1990 and 2010. Your goal is to predict the `Direction` variable, which has two levels (Down and Up). For this task, you should fit the models on a randomly selected subset of your data – the training set – and evaluate their performance on the rest of the data – the test set. I have given you some code below to help you construct these datasets. How accurate are the results compared to simpler methods like logistic regression? Which of these approaches yields the best performance?

```
weekly<-Weekly
set.seed(3) # Set a value for the random number generator to make the results comparable across runs
train <-  sample(nrow(Weekly), 2/3 * nrow(Weekly)) # Randomly select two-thirds of the data
#for sample(whole population, the number of items to choose from)
Weekly_train <- Weekly[train,] # Subset to the training observations
Weekly_test <- Weekly[-train,] # Subset to the test observations

#Logistic regression
glm.fit<- glm(Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume, data=Weekly_train, family = 'binomial')
prob_1<- predict(glm.fit, newdata= Weekly_test, type = 'response')
pred_1<-rep('down', length(prob_1))
pred_1[prob_1>0.5]<-'Up'
table(pred_1, Weekly_test$Direction)
```

```
##
## pred_1 Down  Up
##   down   14  18
##   Up    151 180
```

```r
#the model gives a lot of true up but a lot of false down.
mean(pred_1== Weekly_test$Direction)
```

```
## [1] 0.4958678
```

```r
#Random Forest
library('randomForest')

bag.weekly <-  randomForest(Direction~.-Year-Today,
                            data=Weekly_train,
                            mtry=6)

yhat.bag <-  predict(bag.weekly, newdata=Weekly_test)
table(yhat.bag, Weekly_test$Direction)
```

```
##
## yhat.bag Down  Up
##     Down   69  64
##     Up     96 134
```

```r
mean(yhat.bag != Weekly_test$Direction)
```

```
## [1] 0.4407713
```