Homework 6

Rohan Krishnan

2024-03-22

Problem 1

No submission required

Problem 2

ISLR Chapter 7 Conceptual Exercise 5

(a) As $\lambda \to \infty$, $\hat{g}_2$ is more flexible because it penalizes a higher order of $g(x)$ so it will have the smaller training RSS

(b) As $\lambda \to \infty$, we cannot be certain which function will perform better but there is more of a chance of $\hat{g}_2$ over fitting the data so $\hat{g}_1$ may have a smaller test RSS.

(c) For $\lambda = 0$, there is no penalty so both functions are the same. Therefore, the training and testing RSS will be the same.

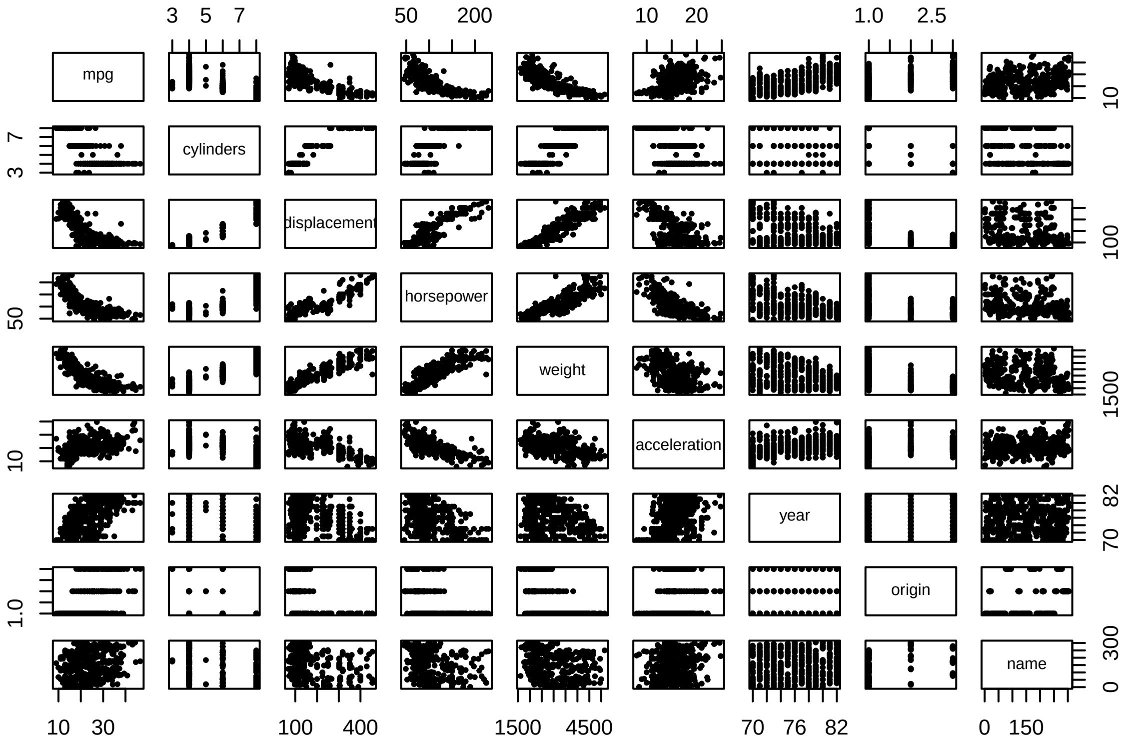
Problem 3

ISLR Chapter 7 Applied Exercise 8

```r
#Load libraries
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(ISLR2)
library(boot)
library(splines)

#Examine relationships
pairs(Auto, cex = 0.4, pch = 19)
```



To focus on a non-linear relationship, I will use weight to predict mpg with 5 models, a regular linear glm, a polynomial, a step wise, and two splines.

```
#GLM
set.seed(50)
fit <- glm(mpg ~ weight, data = Auto)
err <- cv.glm(Auto, fit, K = 10)$delta[1]

#Polynomial
fit.1 <- glm(mpg~poly(weight, 4), data = Auto)
err.1 <- cv.glm(Auto, fit.1, K = 10)$delta[1]

#Step
quants <- quantile(Auto$weight)
Auto$weight_step <- cut(Auto$weight, breaks = quants, include.lowest = TRUE)
fit.2 <- glm(mpg ~ weight_step, data = Auto)
err.2 <- cv.glm(Auto, fit.2, K = 10)$delta[1]

#Regression Spline
fit.3 <- glm(mpg ~ splines::bs(weight, df = 4), data = Auto)
err.3 <- cv.glm(Auto, fit.3, K = 10)$delta[1]

#Natural Spline
fit.4 <- glm(mpg ~ splines::ns(weight, df = 4), data = Auto)
err.4 <- cv.glm(Auto, fit.4, K = 10)$delta[1]

#Compare fits
anova(fit, fit.1, fit.2, fit.3, fit.4)
```

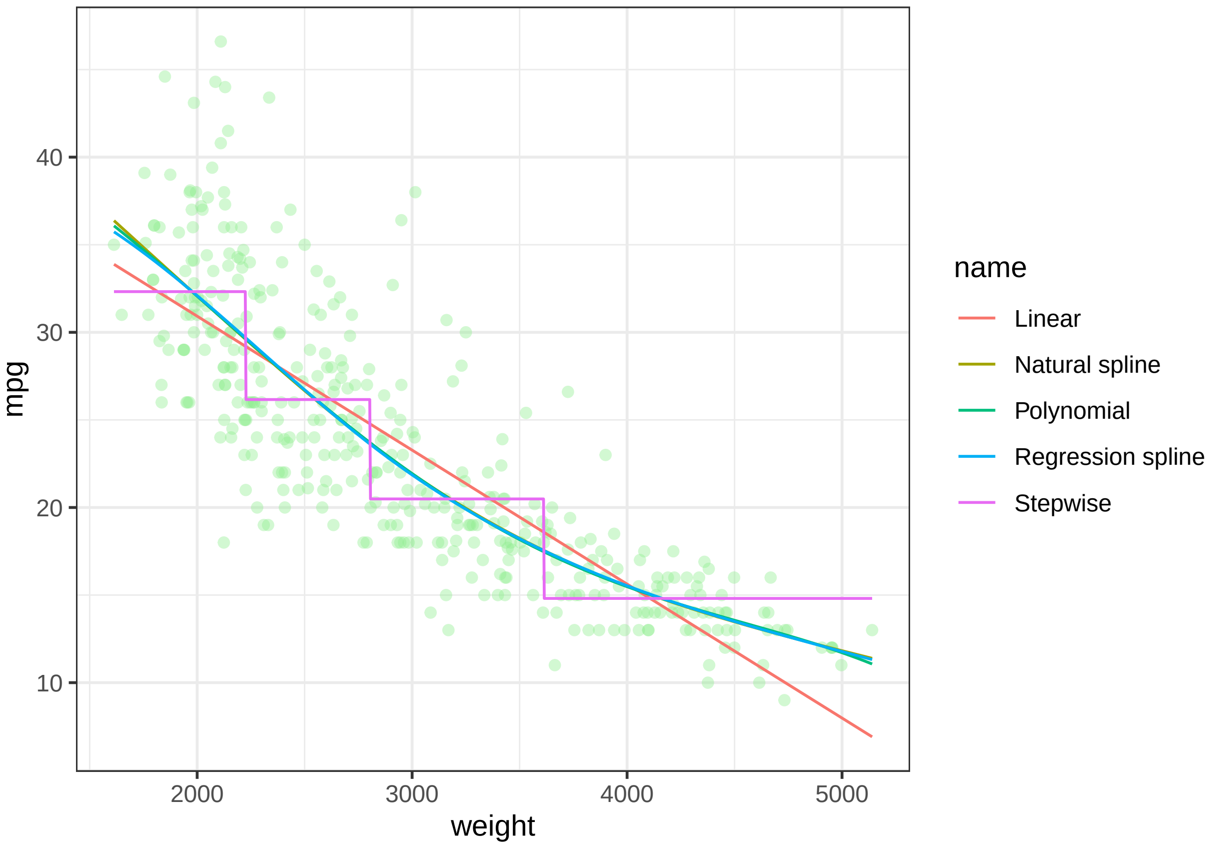
```
## Analysis of Deviance Table
##
```

```
## Model 1: mpg ~ weight
## Model 2: mpg ~ poly(weight, 4)
## Model 3: mpg ~ weight_step
## Model 4: mpg ~ splines::bs(weight, df = 4)
## Model 5: mpg ~ splines::ns(weight, df = 4)
##   Resid. Df Resid. Dev Df Deviance
## 1       390     7321.2
## 2       387     6777.0  3   544.27
## 3       388     7204.2 -1  -427.23
## 4       387     6773.6  1   430.57
## 5       387     6778.6  0    -4.95
```

```
#Display errors
err.all <- c(err, err.1, err.2, err.3, err.4); err.all
```

```
## [1] 18.84161 17.67531 18.69905 17.63980 17.80457
```

```
#Graph
x <- seq(min(Auto$weight), max(Auto$weight), length.out=1000)
pred <- data.frame(
  x = x,
  "Linear" = predict(fit, data.frame(weight = x)),
  "Polynomial" = predict(fit.1, data.frame(weight = x)),
  "Stepwise" = predict(fit.2, data.frame(weight_step = cut(x, breaks = quants, include.lowest = TRUE)))
  "Regression spline" = predict(fit.3, data.frame(weight = x)),
  "Natural spline" = predict(fit.4, data.frame(weight = x)),
  check.names = FALSE
)

pred <- pivot_longer(pred, -x)
ggplot(Auto, aes(weight, mpg)) +
  geom_point(color = alpha("light green", 0.4)) +
  geom_line(data = pred, aes(x, value, color = name)) +
  theme_bw()
```



The regression spline appeared to have the lowest error.

ISLR Chapter 7 Applied Exercise 9

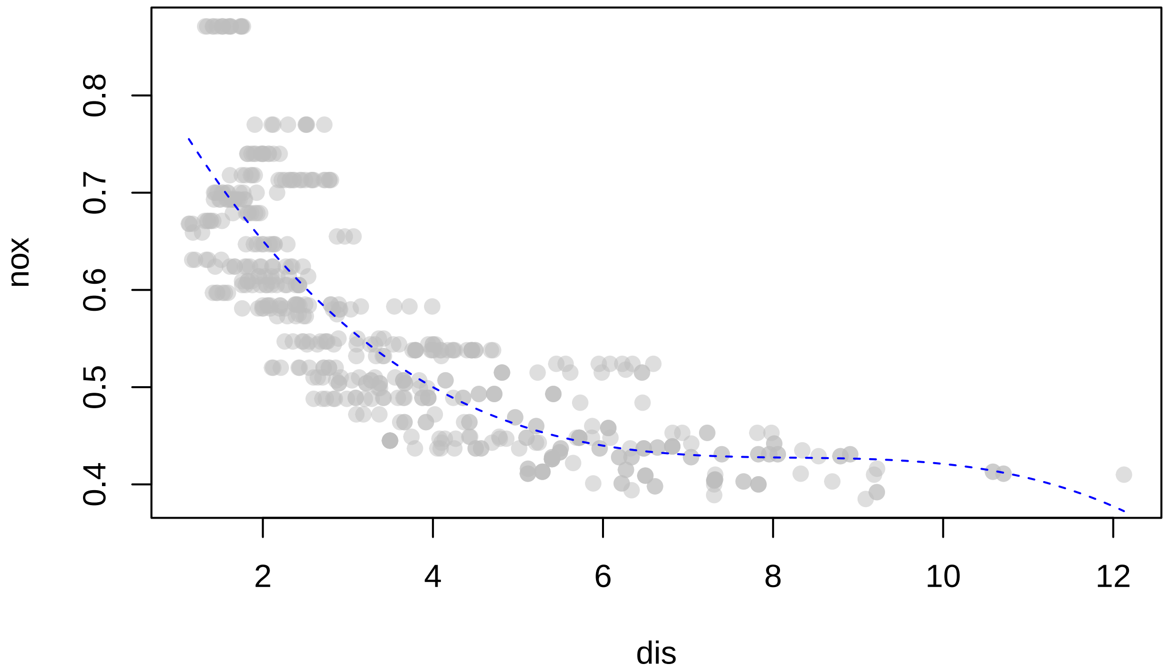
(a)

```
#Polynomial regression
fit.a <- glm(nox ~ poly(dis, 3), data = Boston)
summary(fit.a)
```

```
##
## Call:
## glm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -5.124 4.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.003852802)
##
##     Null deviance: 6.7810  on 505  degrees of freedom
## Residual deviance: 1.9341  on 502  degrees of freedom
## AIC: -1370.9
```

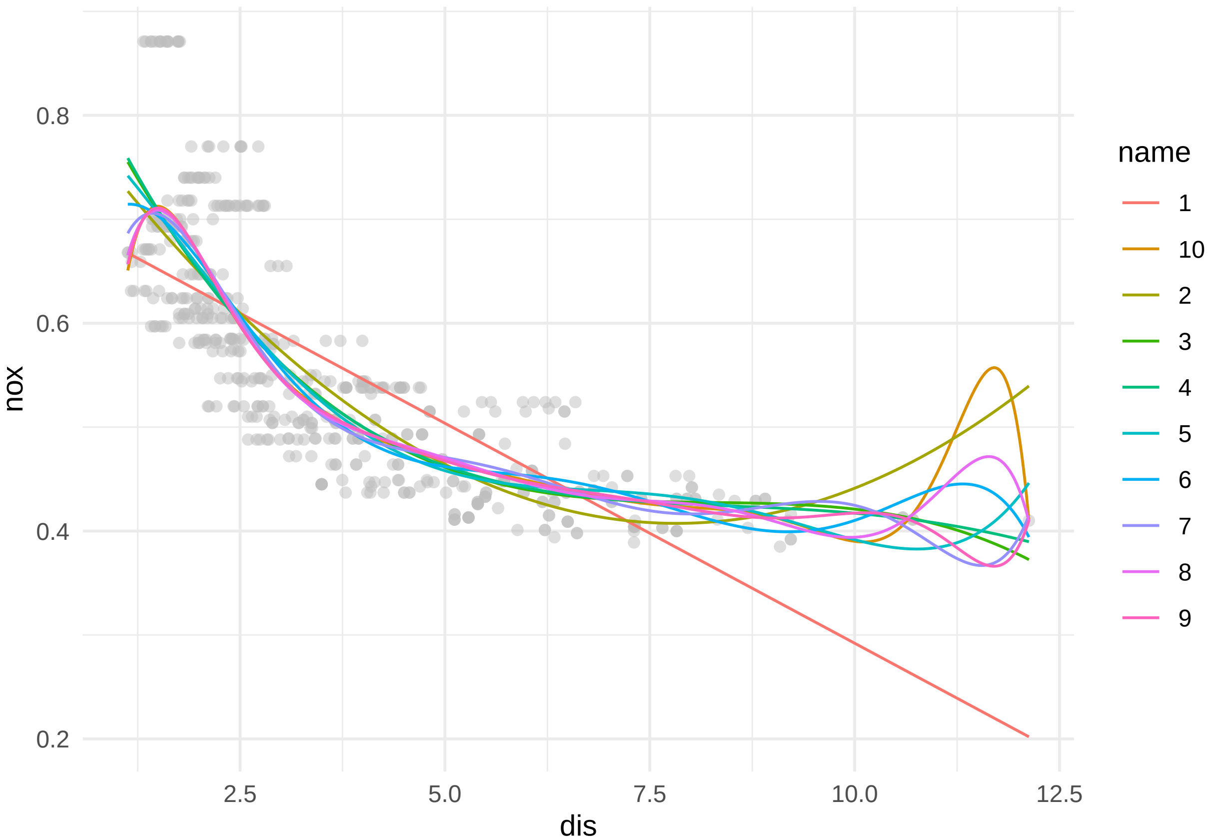
```
##
## Number of Fisher Scoring iterations: 2
```

```
#Plot
plot(nox ~ dis, Boston, col = alpha("grey", 0.5), pch = 19)
lines(seq(min(Boston$dis), max(Boston$dis), length.out = 1000),
      predict(fit.a, data.frame(dis = seq(min(Boston$dis), max(Boston$dis), length.out = 1000))),
              col = "blue", lty = 2)
```



(b)

```
#Generate polynomial fits from 1:10
poly.fits <- lapply(1:10, function(i) glm(nox ~ poly(dis, i), data = Boston))

#Plot
x.axis <- seq(min(Boston$dis), max(Boston$dis), length.out=1000)
pred <- data.frame(lapply(poly.fits, function(a) predict(a, data.frame(dis = x.axis))))
colnames(pred) <- 1:10
pred$x <- x.axis
pred <- pivot_longer(pred, !x)
ggplot(Boston, aes(dis, nox)) +
  geom_point(color = alpha("grey", 0.5)) +
  geom_line(data = pred, aes(x, value, color = name)) +
  theme_minimal()
```




```
#Get RSS
do.call(anova, poly.fits)[,2]
```

```
##  [1] 2.768563 2.035262 1.934107 1.932981 1.915290 1.878257 1.849484 1.835630
##  [9] 1.833331 1.832171
```

(c)

```
#CV to find optimal polynomial
opt.selection <- sapply(1:10, function(i){
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  cv.glm(Boston, fit, K = 10)$delta[1]
})

which.min(opt.selection)
```

```
## [1] 3
```

Based on cross validation, the optimal degress is 3 because higher values being to overfit and increasee the error on the data.

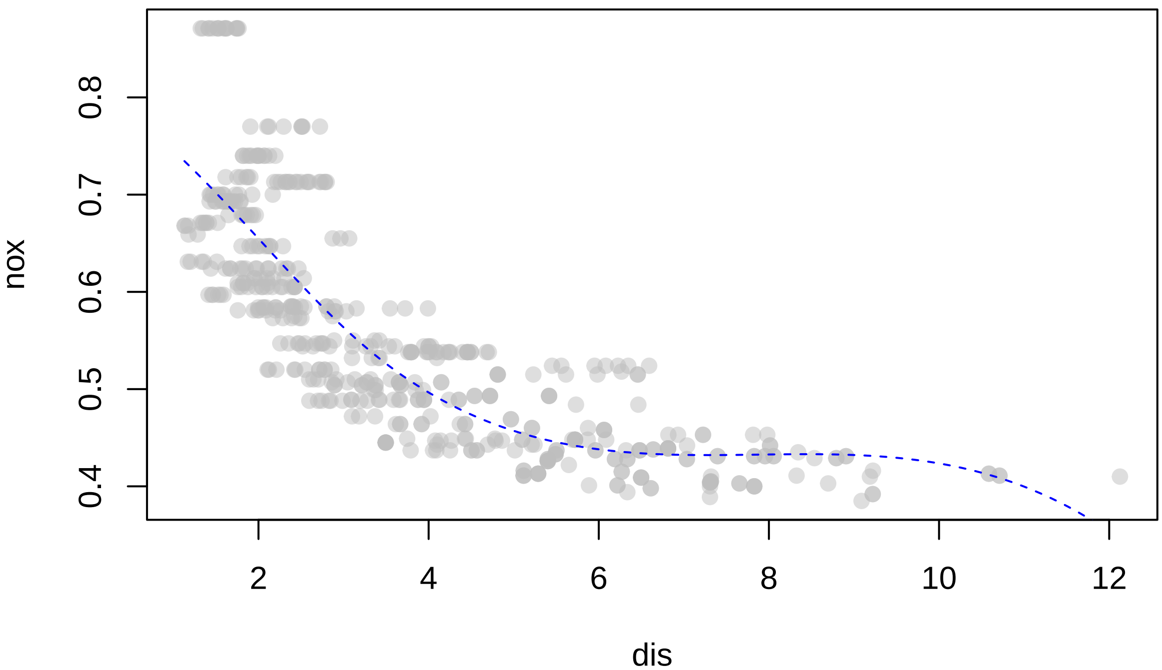
(d)

```
#Fit
fit.d <- glm(nox ~ splines::bs(dis, df = 4), data = Boston)
summary(fit)
```

```
##
## Call:
## glm(formula = mpg ~ weight, data = Auto)
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 46.216524   0.798673   57.87   <2e-16 ***
## weight      -0.007647   0.000258  -29.64   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 18.77239)
##
##     Null deviance: 23819.0  on 391  degrees of freedom
## Residual deviance:  7321.2  on 390  degrees of freedom
## AIC: 2265.9
##
## Number of Fisher Scoring iterations: 2
```

```r
#Plot
plot(nox ~ dis, Boston, col = alpha("grey", 0.50), pch = 19)
lines(seq(min(Boston$dis), max(Boston$dis), length.out = 1000),
      predict(fit.d, data.frame(dis =
                                 seq(min(Boston$dis), max(Boston$dis), length.out = 1000))),
              col = "blue", lty = 2)
```

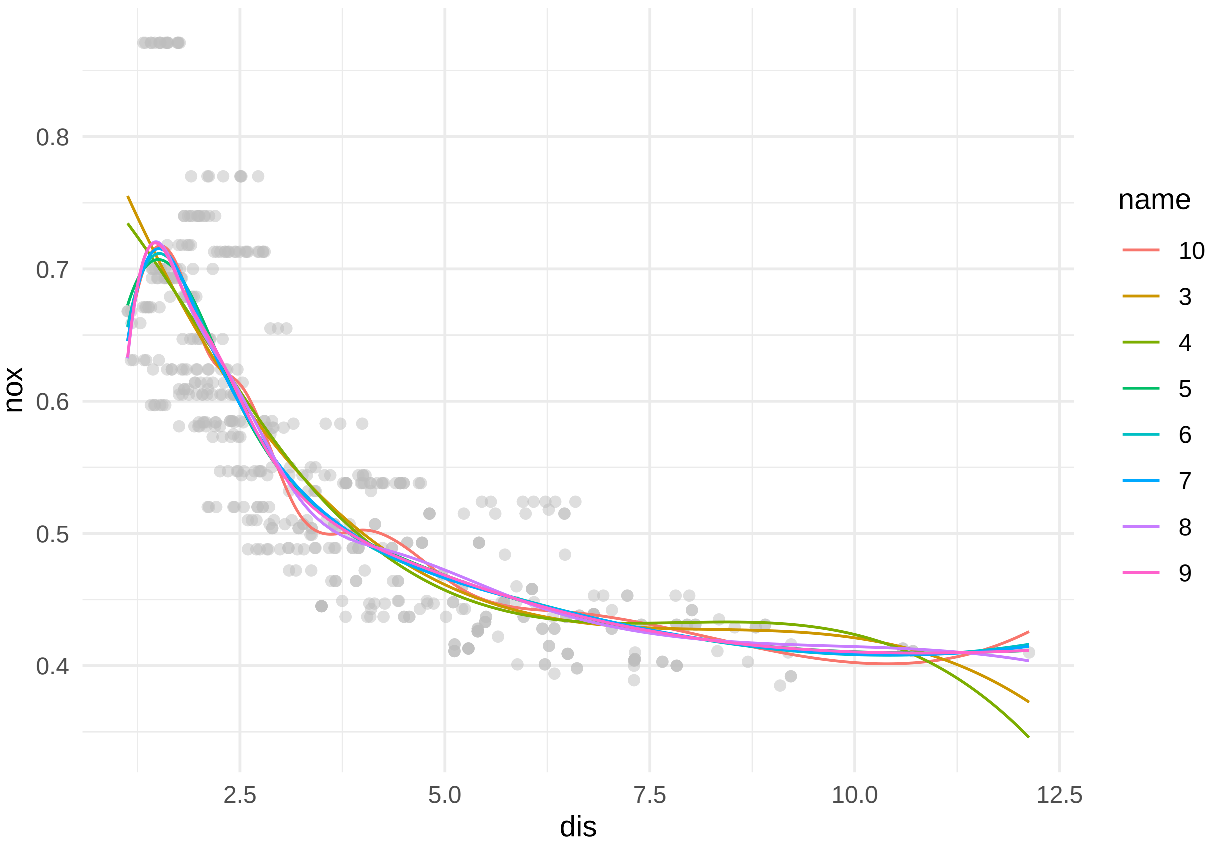


In regression splines, the knots are chosen based on quantiles of the data.

(e)

```r
#Generate fits
spline.fits <- lapply(3:10, function(i){
  glm(nox ~ bs(dis, df = i), data = Boston)
})

#Plot
pred <- data.frame(lapply(spline.fits, function(b) predict(b, data.frame(dis = x.axis))))
colnames(pred) <- 3:10
```

```r
pred$x <- x.axis
pred <- pivot_longer(pred, !x)
ggplot(Boston, aes(dis, nox)) +
  geom_point(color = alpha("grey", 0.50)) +
  geom_line(data = pred, aes(x, value, color = name)) +
  theme_minimal()
```



The highest df splines appear to be beginning to over fitting the data.

(f)

```r
set.seed(52)
opt.spline <- sapply(3:10, function(i){
  fit <- glm(nox ~ splines::bs(dis, df = i), data = Boston)
  cv.glm(Boston, fit, K = 10)$delta[1]
})
which.min(opt.spline)
```

```
## [1] 6
```

Selected 8 degrees of freedom.

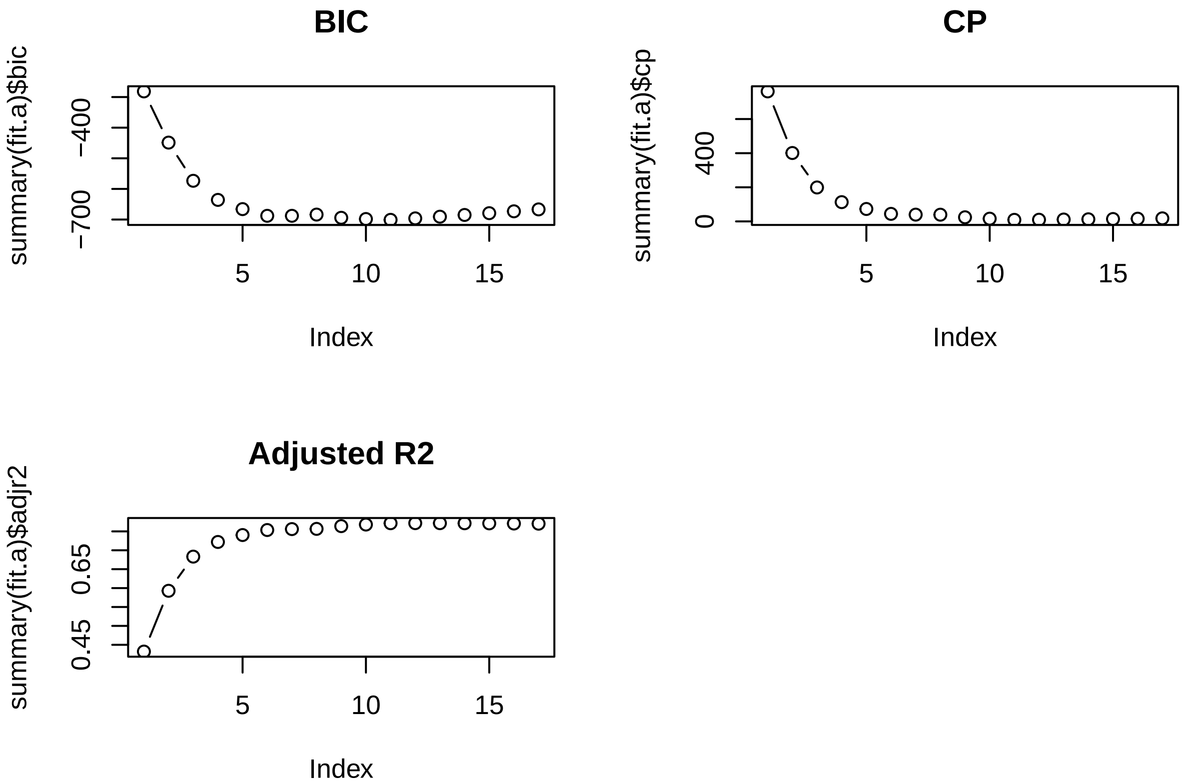
ISLR Chapter 7 Applied Exercise 10

(a)

```r
library(leaps)

set.seed(60)
train <- rep(TRUE, nrow(College))
```

```r
train[sample(1:nrow(College), nrow(College) * 1/3)] <- FALSE
fit.a <- regsubsets(Outstate ~ ., data = College[train,], nvmax = 17, method = "forward")

par(mfrow = c(2,2))
plot(summary(fit.a)$bic, type = "b", main = "BIC")
plot(summary(fit.a)$cp, type = "b", main = "CP")
plot(summary(fit.a)$adjr2, type = "b", main = "Adjusted R2")

coef(fit.a, id = 6)
```

```
##   (Intercept)    PrivateYes    Room.Board      Terminal   perc.alumni 
## -4755.7850866  2851.4882227     0.9795393    43.5895940    48.3939096 
##        Expend     Grad.Rate 
##     0.2115436    33.5099579 
```



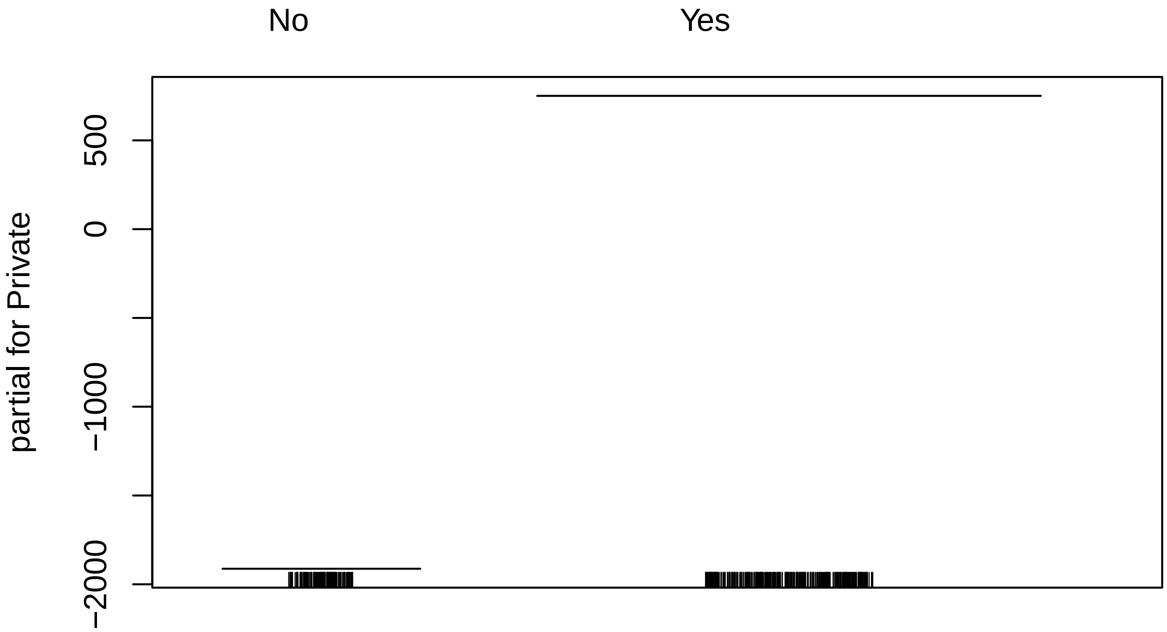
Pick 6 since CP and BIC slightly increase after 6.

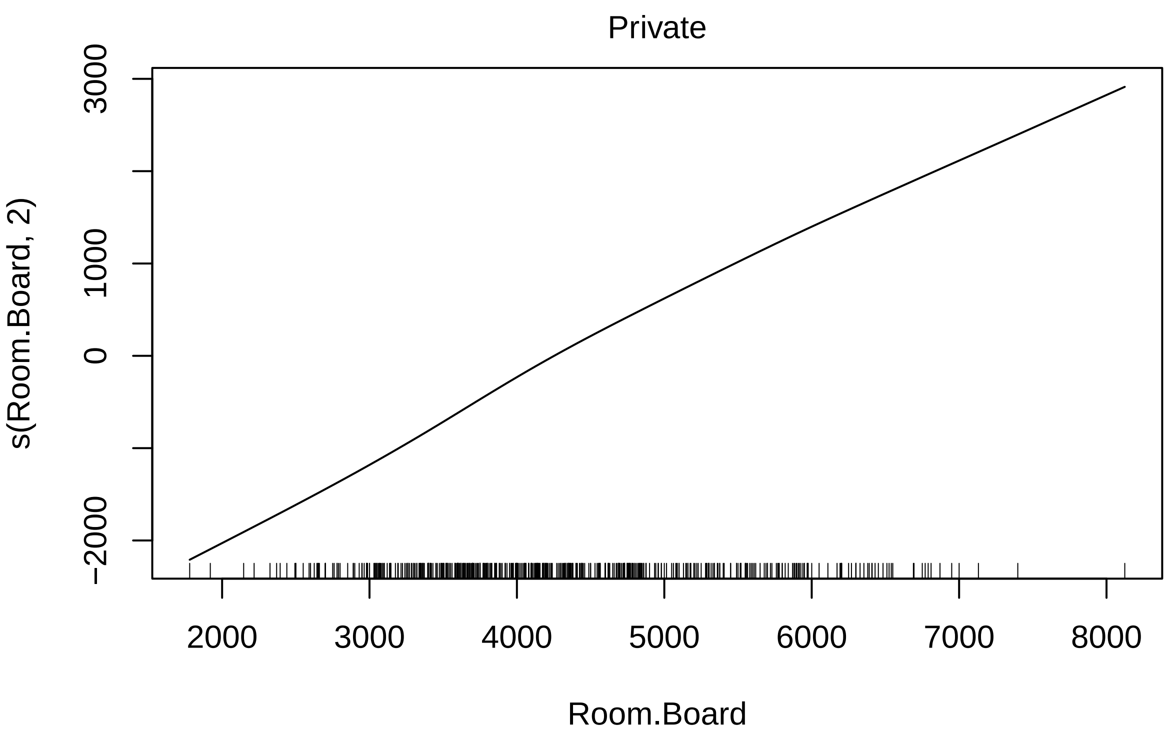
(b)

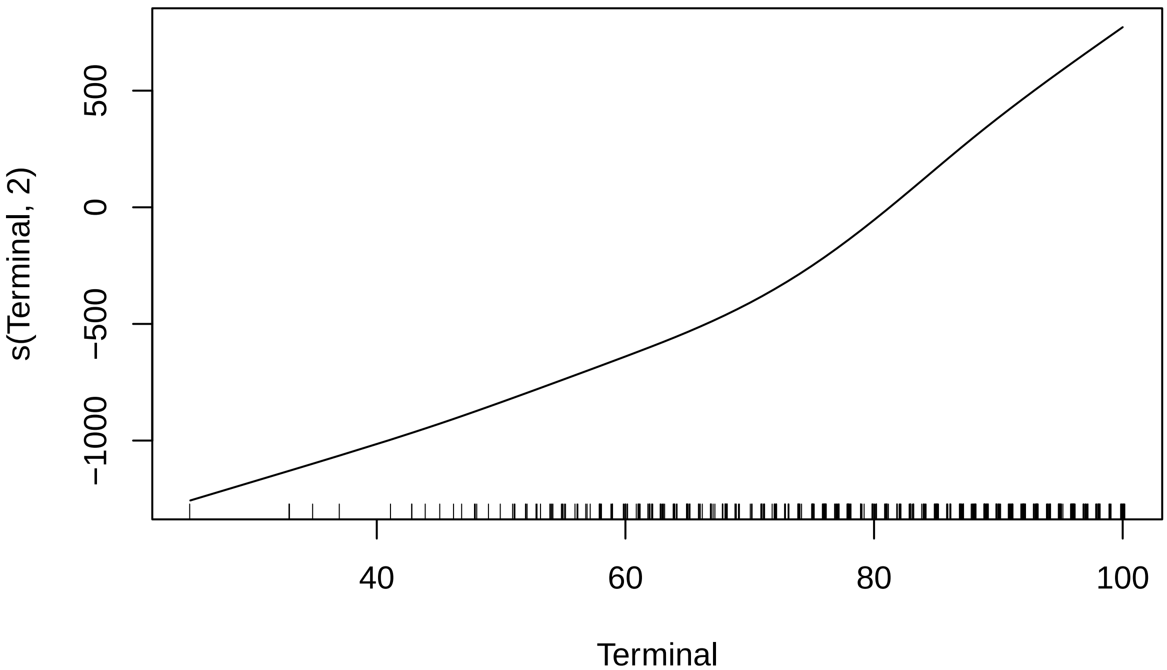
```r
library(gam)
```

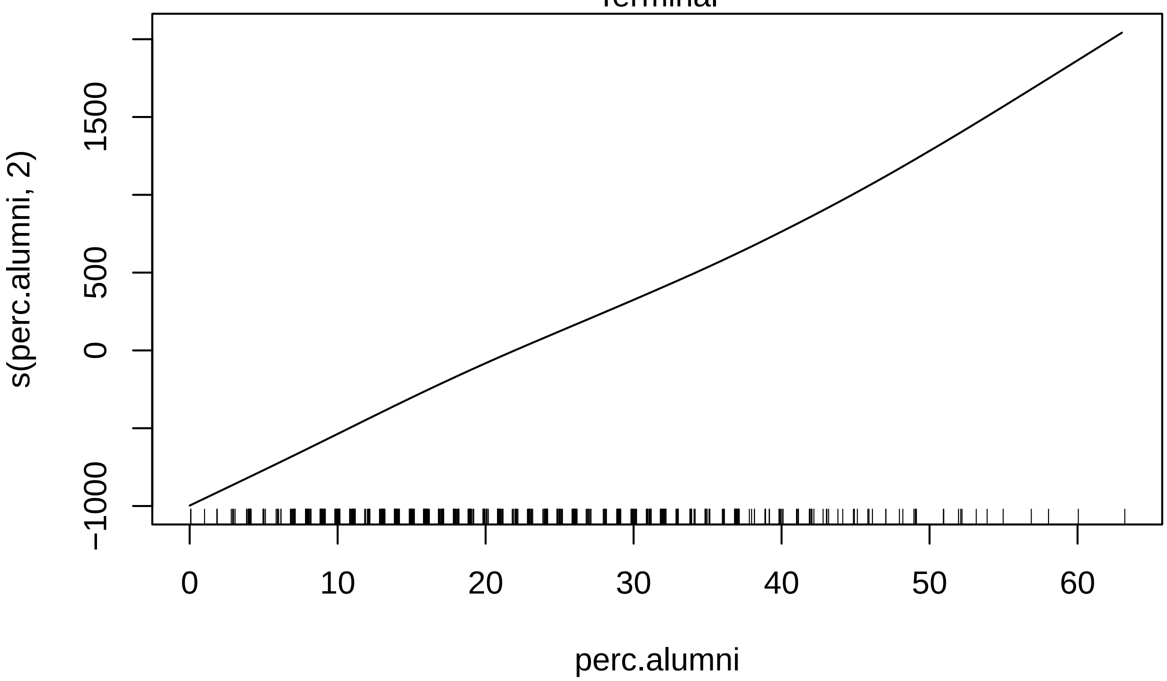
```
## Loading required package: foreach
##
## Attaching package: 'foreach'
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
## Loaded gam 1.22-3
```

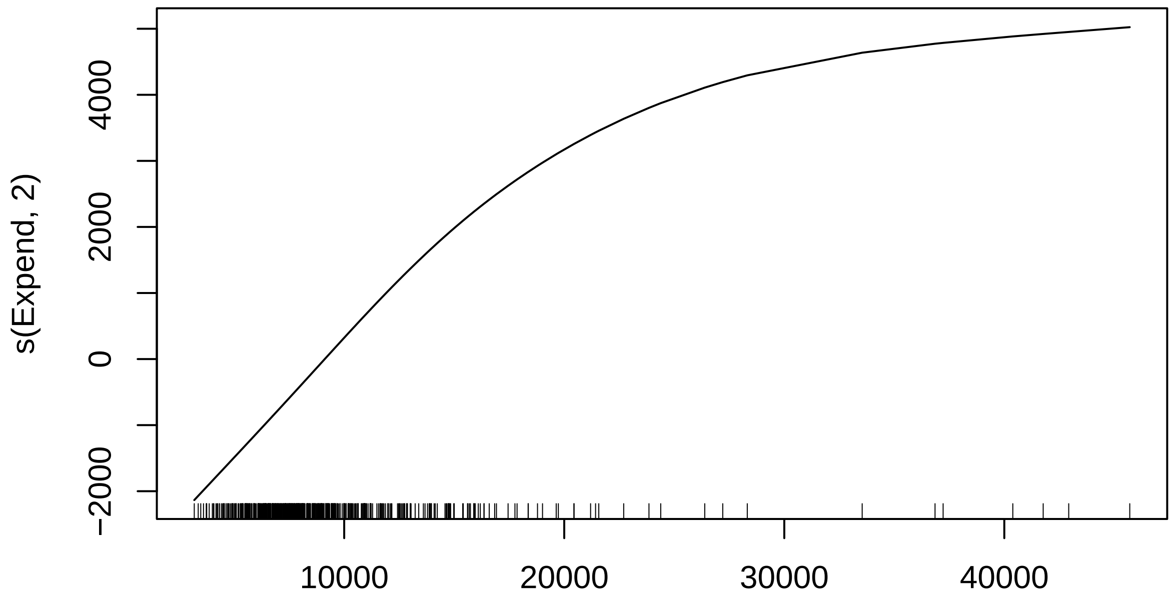
```
fit.b <- gam(Outstate ~ Private + s(Room.Board, 2) + s(Terminal, 2) + s(perc.alumni,2) +
               s(Expend,2) + s(Grad.Rate, 2), data = College[train,])

plot(fit.b)
```

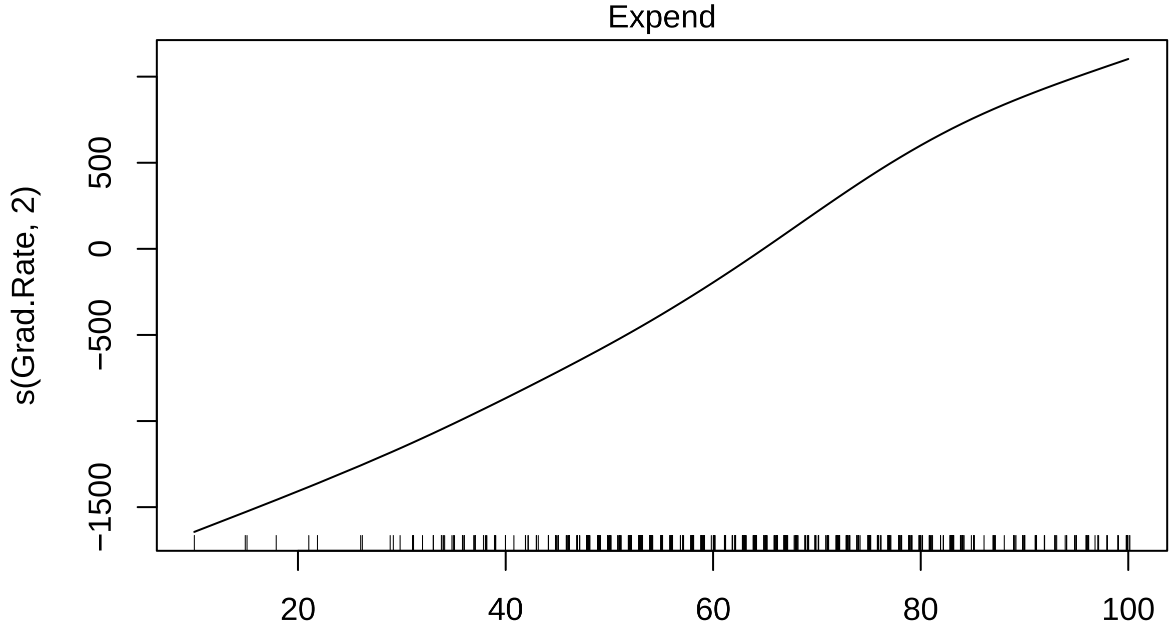





s(Terminal, 2)
500
0
−500
−1000
40
60
80
100
Terminal


s(perc.alumni, 2)
1500
500
0
−1000
0
10
20
30
40
50
60
perc.alumni





There appears to be one non linear variables but the other variables look linear.

(c)

```
pred <- predict(fit.b, College[!train,])
err.gam <- mean((College$Outstate[!train] - pred)^2)
1 - err.gam / mean((College$Outstate[!train] - mean(College$Outstate[!train]))^2)
```

```
## [1] 0.7630411
```

The error appears quite large though that could be because of the units of tuition. The R2 is around 0.76 which is quite good.

(d)

```r
summary(fit.b)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board, 2) + s(Terminal,
##     2) + s(perc.alumni, 2) + s(Expend, 2) + s(Grad.Rate, 2),
##     data = College[train, ])
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7044.54 -1188.18    42.18  1271.43  5325.89
##
## (Dispersion Parameter for gaussian family taken to be 3603153)
##
##     Null Deviance: 8384825614 on 517 degrees of freedom
## Residual Deviance: 1823196951 on 506.0004 degrees of freedom
## AIC: 9304.29
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                   Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private            1 2378689692 2378689692 660.169 < 2.2e-16 ***
## s(Room.Board, 2)   1 1865168713 1865168713 517.649 < 2.2e-16 ***
## s(Terminal, 2)     1  619929691  619929691 172.052 < 2.2e-16 ***
## s(perc.alumni, 2)  1  383391372  383391372 106.404 < 2.2e-16 ***
## s(Expend, 2)       1  517741577  517741577 143.691 < 2.2e-16 ***
## s(Grad.Rate, 2)    1  123968707  123968707  34.406 8.089e-09 ***
## Residuals        506 1823196951    3603153
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                  Npar Df Npar F     Pr(F)
## (Intercept)
## Private
## s(Room.Board, 2)        1  1.831   0.17659
## s(Terminal, 2)          1  3.122   0.07785 .
## s(perc.alumni, 2)       1  0.753   0.38589
## s(Expend, 2)            1 48.703 9.399e-12 ***
## s(Grad.Rate, 2)         1  3.523   0.06110 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There appears to be significant evidence of a non-linear relationship for Expend.