GAM Homework

Tao He

2/14/2022

7.3. Suppose we fit a curve with basis functions b1(X) = X, b2(X) = (X − 1)^2I(X   1). (Note that I(X   1) equals 1 for X   1 and 0 otherwise.) We fit the linear regression model

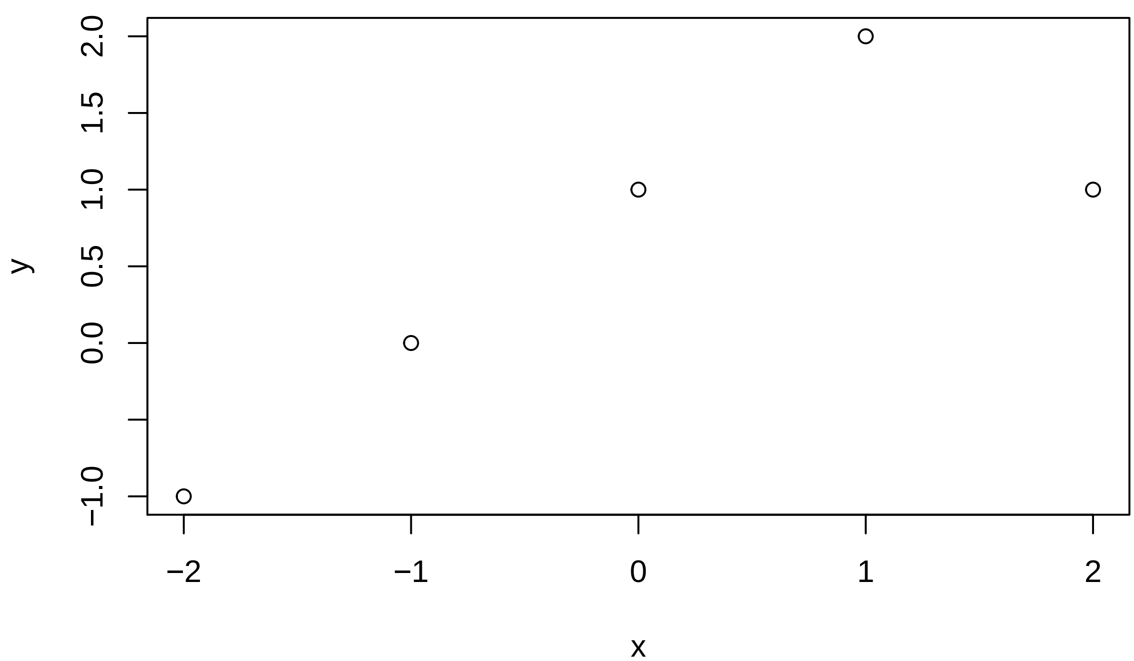
$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$$

, and obtain coefficient estimates

$$\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = -2$$

. Sketch the estimated curve between X = −2 and X = 2. Note the intercepts, slopes, and other relevant information.

```
x <- -2:2
y <- 1 + x + (-2) * (x - 1)^2 * I(x > 1)
plot(x, y)
```



When x is in [-2, 1), the curve is linear with the intercept(1) and slope(x), y = 1+x, and when x is in [1,2], the curve is quadratic and y = 1+x-2(x-1)^2.

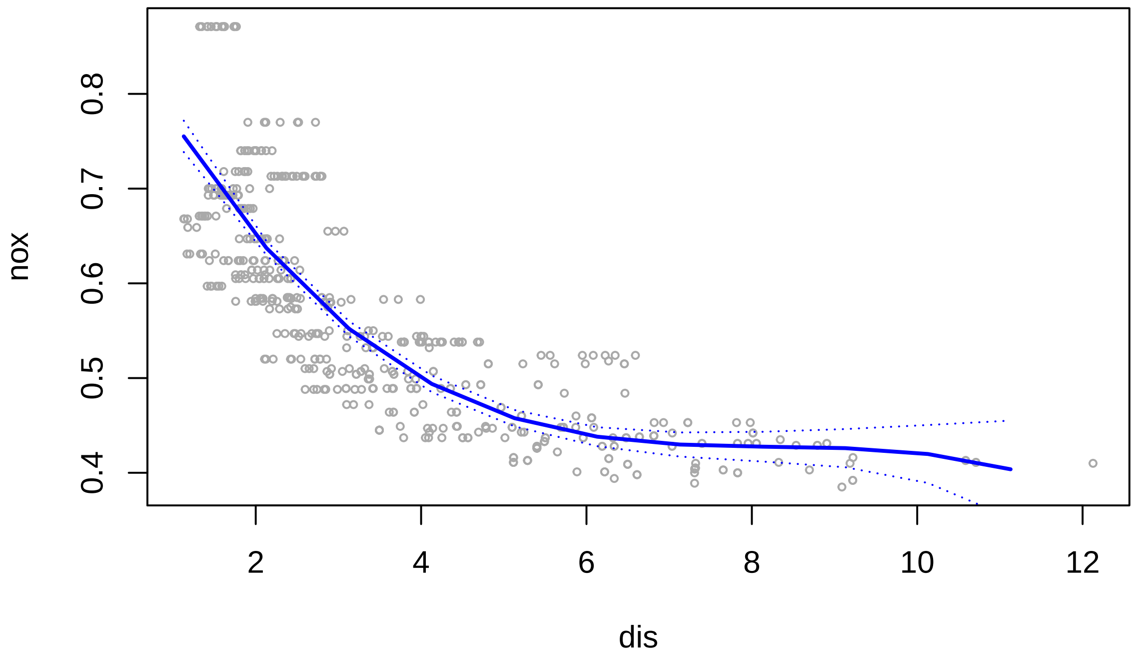
7.9. This question uses the variables dis (the weighted mean of distances to five Boston employment centers) and nox (nitrogen oxides concentration in parts per 10 million) from the Boston data. We will treat dis as the predictor and nox as the response.

(a) Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

```r
set.seed(679)
fit <- lm(nox ~ poly(dis, 3), data = Boston)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## 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
##
## Residual standard error: 0.06207 on 502 degrees of freedom
## Multiple R-squared:  0.7148, Adjusted R-squared:  0.7131
## F-statistic: 419.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

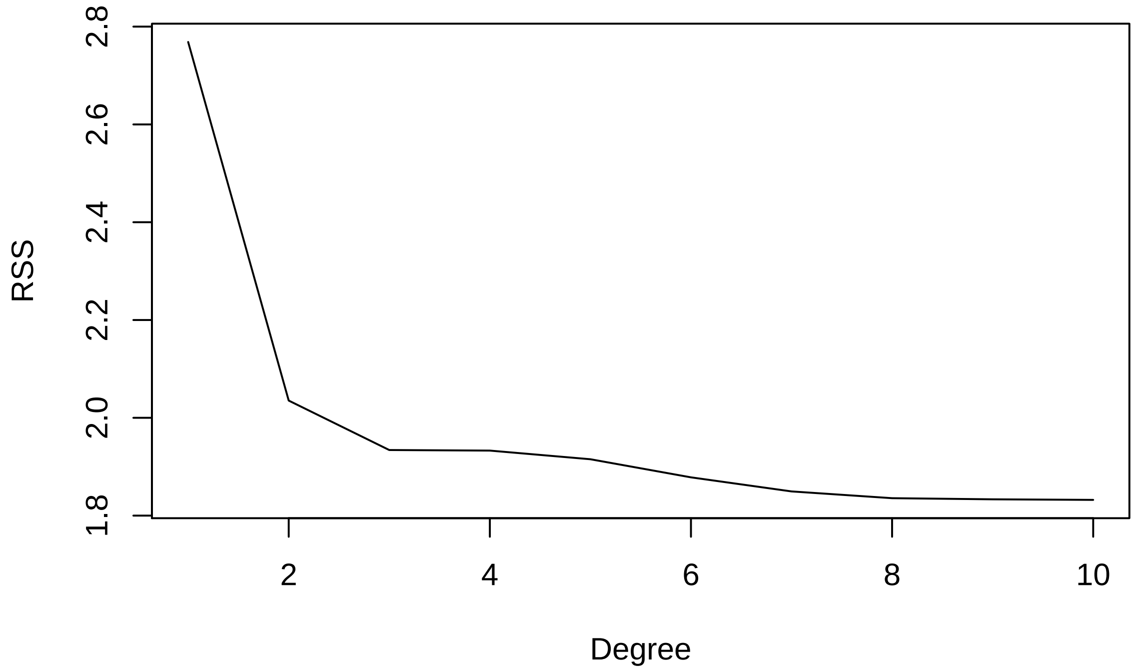
```r
dislims <- range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2])
preds <- predict(fit, newdata = list(dis = dis.grid),
se = TRUE)
se.bands <- cbind(preds$fit + 2 * preds$se.fit,
preds$fit - 2 * preds$se.fit)

plot(nox ~ dis, data = Boston, xlim = dislims, cex = .5, col = "darkgrey")
lines(dis.grid, preds$fit, lwd = 2, col = "blue")
matlines(dis.grid, se.bands, lwd = 1, col = "blue", lty = 3)
```



We can say that all polynomial terms are significant.

(b) Plot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.

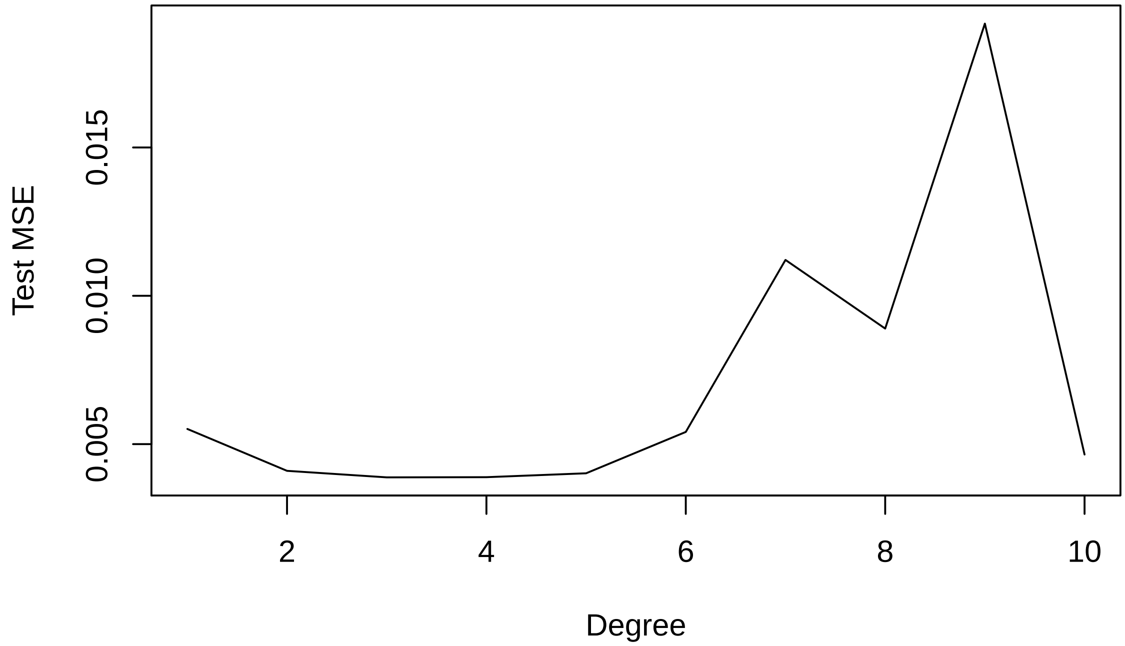
```
rss <- rep(NA, 10)

for (i in 1:10){
  fit <- lm(nox ~ poly(dis, i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}

plot(1:10, rss, xlab = "Degree", ylab = "RSS", type = "l")
```



It shows the RSS decreases when the degree of polynomial increases.

(c) Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.

```
deltas <- rep(NA, 10)
for (i in 1:10){
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  deltas[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}

plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
```

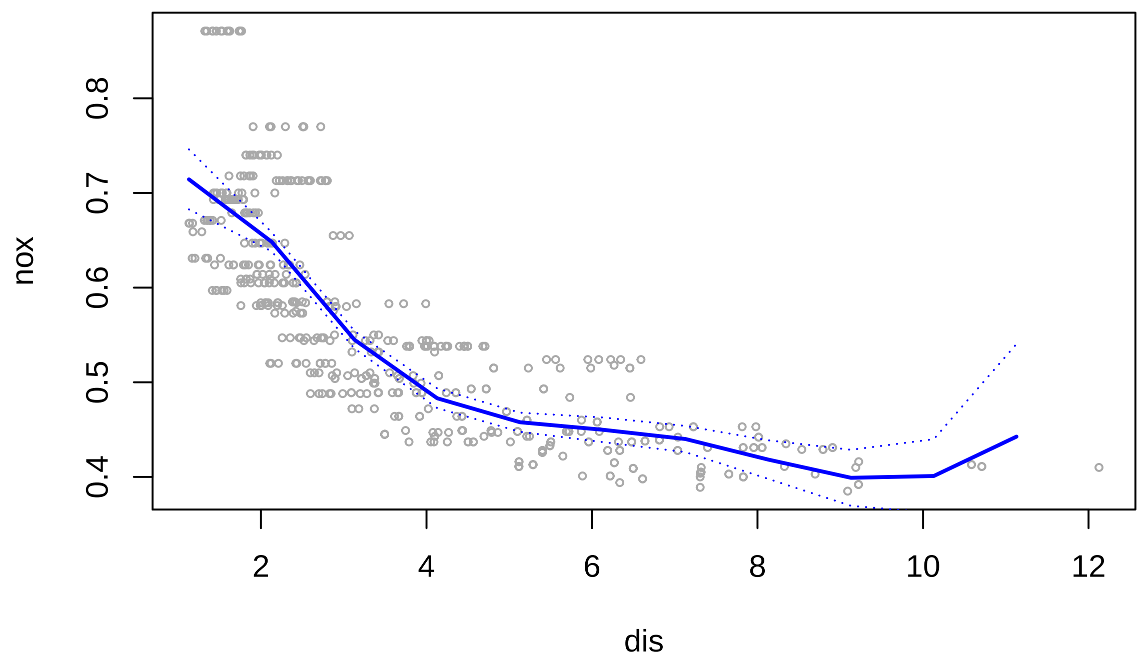


(d) Use the bs() function to fit a regression spline to predict nox using dis. Report the output for the fit using four degrees of freedom. How did you choose the knots? Plot the resulting fit.

```
fit <- lm(nox ~ bs(dis, knots = c(3, 7, 12)), data = Boston)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, knots = c(3, 7, 12)), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.130710 -0.039850 -0.008357  0.027792  0.188518
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   0.714346   0.015846  45.081  < 2e-16 ***
## bs(dis, knots = c(3, 7, 12))1 -0.006626   0.024307  -0.273    0.785
## bs(dis, knots = c(3, 7, 12))2 -0.296980   0.018293 -16.234  < 2e-16 ***
## bs(dis, knots = c(3, 7, 12))3 -0.215329   0.037539  -5.736 1.68e-08 ***
## bs(dis, knots = c(3, 7, 12))4 -0.400490   0.050090  -7.995 9.06e-15 ***
## bs(dis, knots = c(3, 7, 12))5 -0.177922   0.116378  -1.529    0.127
## bs(dis, knots = c(3, 7, 12))6 -0.304346   0.063378  -4.802 2.08e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06137 on 499 degrees of freedom
## Multiple R-squared:  0.7229, Adjusted R-squared:  0.7196
## F-statistic:   217 on 6 and 499 DF,  p-value: < 2.2e-16
```

```
dislims <- range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2])
preds <- predict(fit, newdata = list(dis = dis.grid),
se = TRUE)
se.bands <- cbind(preds$fit + 2 * preds$se.fit,
preds$fit - 2 * preds$se.fit)

plot(nox ~ dis, data = Boston, xlim = dislims, cex = .5, col = "darkgrey")
lines(dis.grid, preds$fit, lwd = 2, col = "blue")
matlines(dis.grid, se.bands, lwd = 1, col = "blue", lty = 3)
```

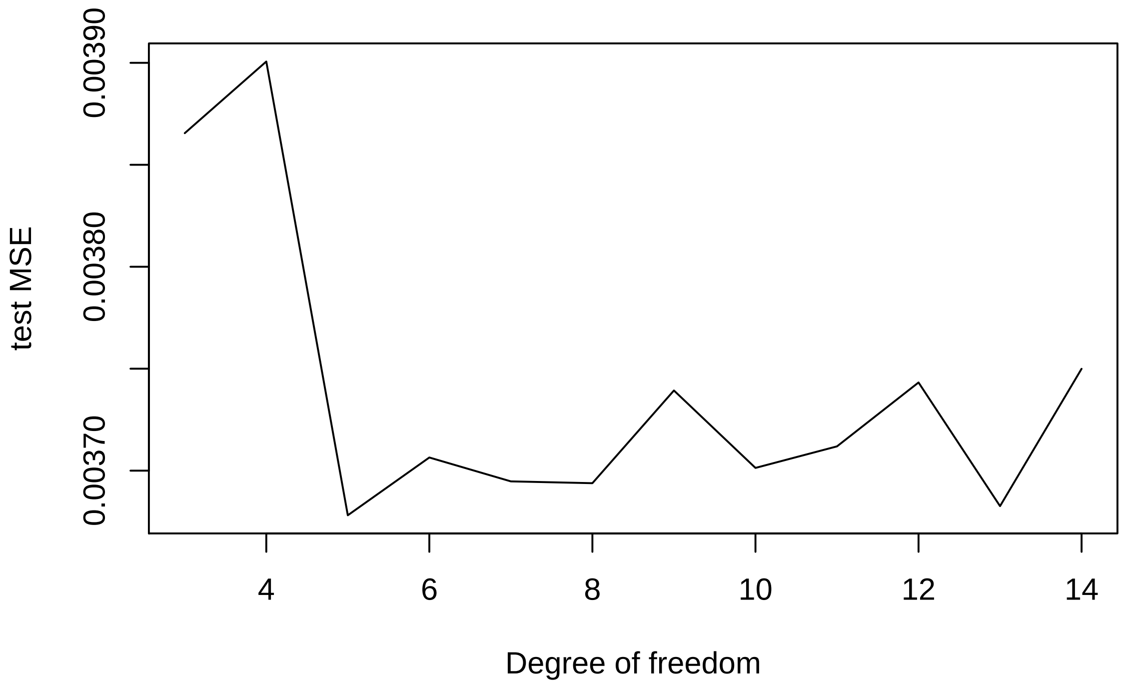


(e) Now fit a regression spline for a range of degrees of freedom, and plot the resulting fits and report the resulting RSS. Describe the results obtained.

```
rss <- rep(NA, 14)
for (i in 3:14){
  fit <- lm(nox ~ bs(dis, df = i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}

plot(3:14, rss[-c(1,2)], xlab = "Degrees of freedom", ylab = "RSS", type = "l")
```

(f) Perform cross-validation or another approach in order to select the best degrees of freedom for a regression spline on this data. Describe your results.

```
cv <- rep(NA, 14)
for (i in 3:14){
  fit <- glm(nox ~ bs(dis, df = i), data = Boston)
  cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}

plot(3:14, cv[-c(1,2)], xlab = "Degree of freedom", ylab = "test MSE", type = "l")
```



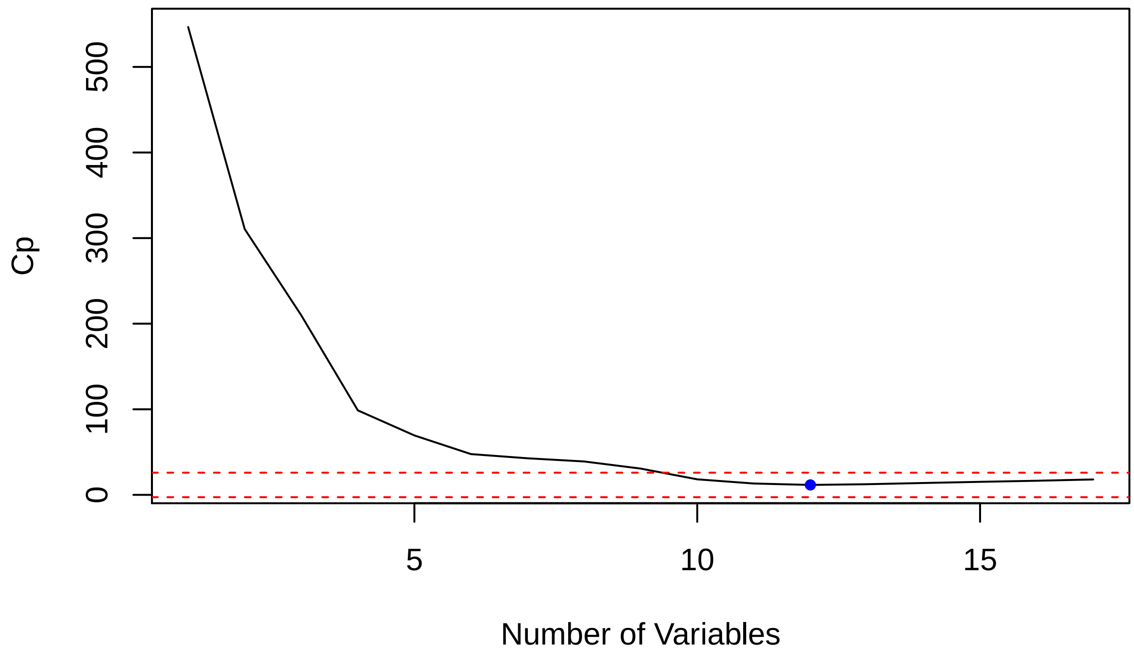
Test MSE is minimum for 13 degrees of freedom.

7.10. This question relates to the College data set.

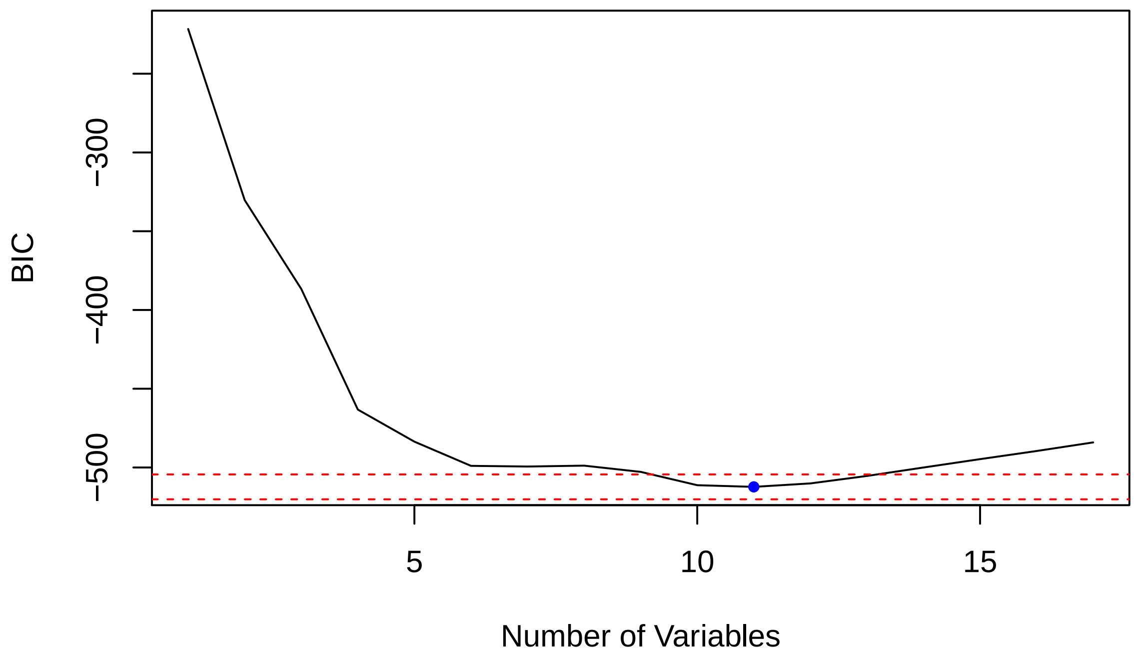
(a) Split the data into a training set and a test set.Using out-of-state tuition as the response and the other variables as the predictors, perform forward step-wise selection on the training set in order to identify a satisfactory model that uses just a subset of the predictors.

```
set.seed(679)
attach(College)

train <- sample(length(Outstate), length(Outstate)/2)
test <- -train

College.train <- College[train,]
College.test <- College[test,]

fit <- regsubsets(Outstate ~., data = College.train, nvmax = 17, method = "forward")
fit.summary <- summary(fit)

# Cp
plot(fit.summary$cp, xlab = "Number of Variables", ylab = "Cp", type = "l")
cp.min <- which.min(fit.summary$cp)
cp.sd <- sd(fit.summary$cp)
points(cp.min, fit.summary$cp[cp.min], col = 'blue', pch = 20)
```

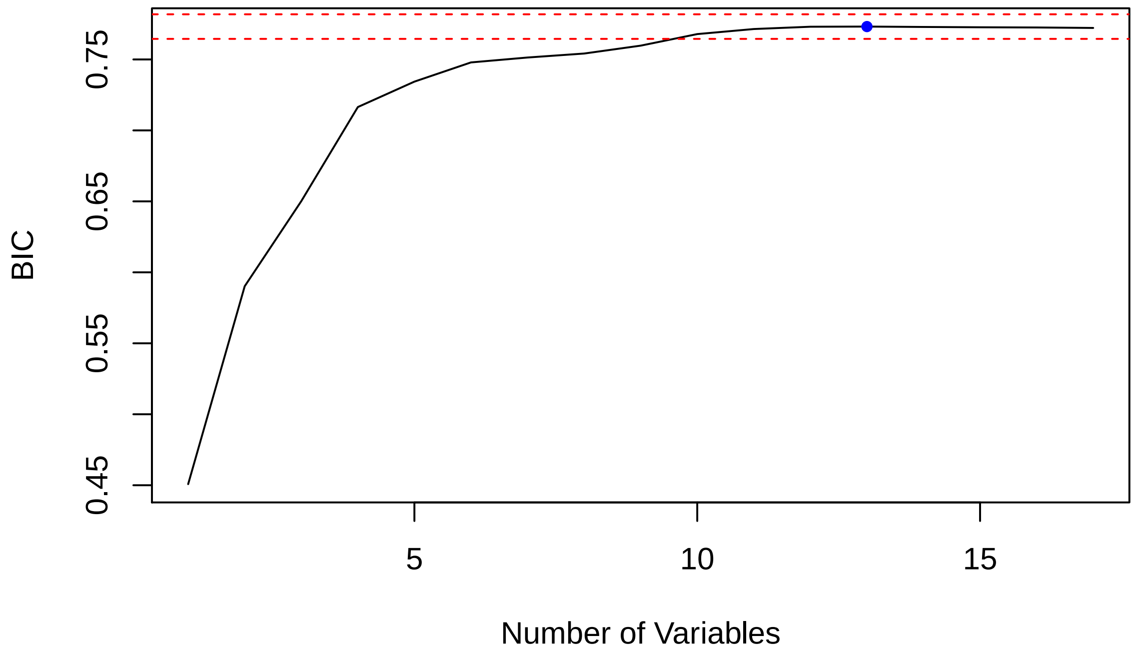
```
abline(h = fit.summary$cp[cp.min] + 0.1 * cp.sd, col = "red", lty= 2)
abline(h = fit.summary$cp[cp.min] - 0.1 * cp.sd, col = "red", lty= 2)
```




```
# BIC
plot(fit.summary$bic, xlab = "Number of Variables", ylab = "BIC", type = "l")
bic.min <- which.min(fit.summary$bic)
bic.sd <- sd(fit.summary$bic)
points(bic.min, fit.summary$bic[bic.min], col = 'blue', pch = 20)
abline(h = fit.summary$bic[bic.min] + 0.1 * bic.sd, col = "red", lty= 2)
abline(h = fit.summary$bic[bic.min] - 0.1 * bic.sd, col = "red", lty= 2)
```



```
# Adjusted R^2
plot(fit.summary$adjr2, xlab = "Number of Variables", ylab = "BIC", type = "l")
adjr2.max <- which.max(fit.summary$adjr2)
adjr2.sd <- sd(fit.summary$adjr2)
points(adjr2.max, fit.summary$adjr2[adjr2.max], col = 'blue', pch = 20)
abline(h = fit.summary$adjr2[adjr2.max] + 0.1 * adjr2.sd, col = "red", lty= 2)
abline(h = fit.summary$adjr2[adjr2.max] - 0.1 * adjr2.sd, col = "red", lty= 2)
```

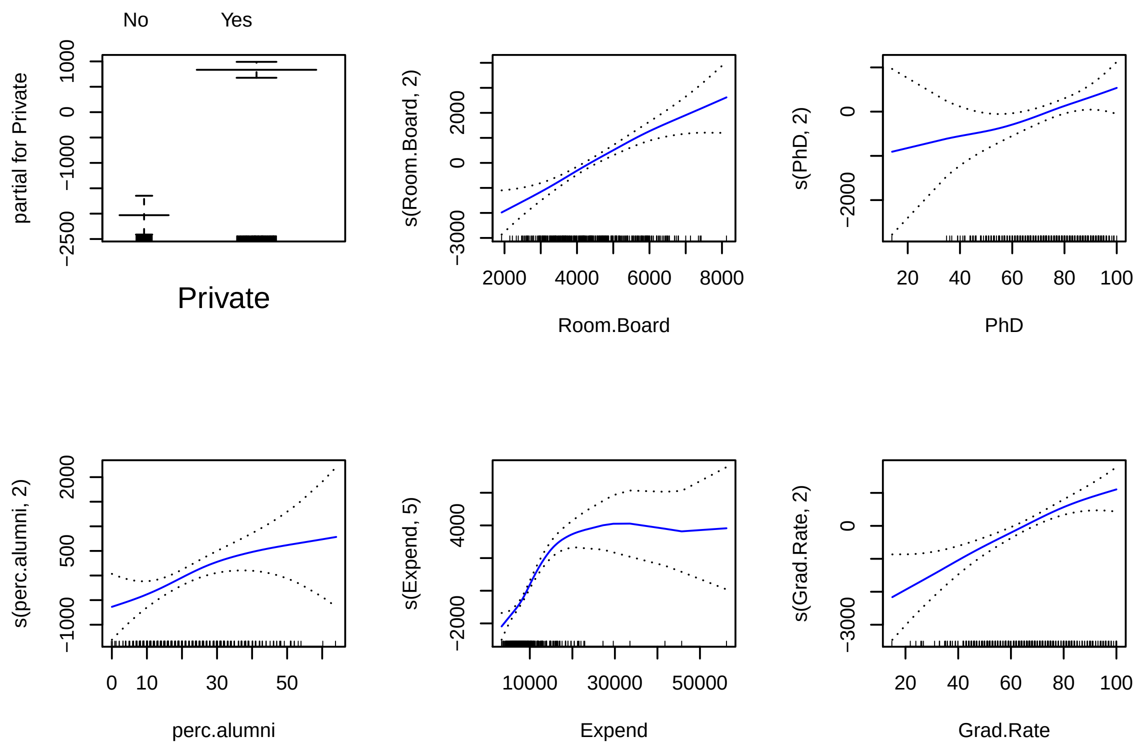



```
fit <- regsubsets(Outstate ~., data = College, method = "forward")
coefficients <- coef(fit, id = 6)
names(coefficients)
```

```
## [1] "(Intercept)" "PrivateYes"  "Room.Board"  "PhD"         "perc.alumni"
## [6] "Expend"      "Grad.Rate"
```

(b) Fit a GAM on the training data, using out-of-state tuition as the response and the features selected in the previous step as the predictors. Plot the results, and explain your findings.

```
fit <- gam(Outstate ~ Private + s(Room.Board, 2) + s(PhD, 2) + s(perc.alumni, 2)
           + s(Expend, 5) + s(Grad.Rate, 2), data=College.train)
par(mfrow = c(2, 3))
plot(fit, se = T, col = "blue")
```



(c) Evaluate the model obtained on the test set, and explain the results obtained.

```
preds <- predict(fit, College.test)
err <- mean((College.test$Outstate - preds)^2)
err
```

```
## [1] 3592242
```

```
tss <- mean((College.test$Outstate - mean(College.test$Outstate))^2)
rss <- 1 - err / tss
rss
```

```
## [1] 0.7799617
```

(d) For which variables, if any, is there evidence of a non-linear relationship with the response?

```
summary(fit)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board, 2) + s(PhD,
##     2) + s(perc.alumni, 2) + s(Expend, 5) + s(Grad.Rate, 2),
##     data = College.train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6673.36 -1065.89   -23.31  1312.80  4973.78
##
## (Dispersion Parameter for gaussian family taken to be 3585962)
##
##     Null Deviance: 6203187949 on 387 degrees of freedom
## Residual Deviance: 1337563897 on 373 degrees of freedom
## AIC: 6973.703
##
## Number of Local Scoring Iterations: 2
```

```
##
## Anova for Parametric Effects
##                     Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private              1 1947681321 1947681321 543.140 < 2.2e-16 ***
## s(Room.Board, 2)     1 1191516451 1191516451 332.272 < 2.2e-16 ***
## s(PhD, 2)            1  367544316  367544316 102.495 < 2.2e-16 ***
## s(perc.alumni, 2)    1  190769391  190769391  53.199 1.823e-12 ***
## s(Expend, 5)         1  401435810  401435810 111.947 < 2.2e-16 ***
## s(Grad.Rate, 2)      1   98144162   98144162  27.369 2.810e-07 ***
## Residuals          373 1337563897    3585962
## ---
## 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  1.1559    0.2830
## s(PhD, 2)                1  0.5126    0.4744
## s(perc.alumni, 2)        1  1.3446    0.2470
## s(Expend, 5)             4 12.8901 7.701e-10 ***
## s(Grad.Rate, 2)          1  1.2846    0.2578
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

7.11. In Section 7.7, it was mentioned that GAMs are generally fit using a backfitting approach. The idea behind back fitting is actually quite simple. We will now explore backfitting in the context of multiple linear regression. Suppose that we would like to perform multiple linear regression, but we do not have software to do so. Instead, we only have software to perform simple linear regression. Therefore, we take the following iterative approach: we repeatedly hold all but one coefficient esti- mate fixed at its current value, and update only that coefficient estimate using a simple linear regression. The process is continued un- til convergence—that is, until the coefficient estimates stop changing.

We now try this out on a toy example.

(a) Generate a response Y and two predictors X1 and X2, with n = 100.

```
set.seed(679)
y <- rnorm(100)
x1 <- rnorm(100)
x2 <- rnorm(100)
```

(b) Initialize

$$\hat{\beta}_1$$

to take on a value of your choice. It does not matter what value you choose.

```
beta1 <- 2
```

(c) Keeping

$$\hat{\beta}_1$$

fixed, fit the model

$$Y - \hat{\beta}_1 X_1 = \hat{\beta}_0 + \hat{\beta}_2 X_2 + \epsilon$$

. You can do this as follows:

```
a <- y - beta1 * x1
beta2 <- lm(a~x2)$coef[2]
```

(d) Keeping

$$\hat{\beta}_2$$

fixed, fit the model

$$Y - \hat{\beta}_2 X_2 = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \epsilon$$

. You can do this as follows:

```
a <- y - beta2 * x2
beta1 <- lm(a~x1)$coef[2]
```

(e) Write a for loop to repeat (c) and (d) 1,000 times. Report the estimates of

$$\hat{\beta}_0$$

,

$$\hat{\beta}_1$$

, and

$$\hat{\beta}_2$$

at each iteration of the for loop. Create a plot in which each of these values is displayed, with

$$\hat{\beta}_0$$

,

$$\hat{\beta}_1$$

, and

$$\hat{\beta}_2$$

each shown in a different color.