ISL Chapter 7 Exercises

Jonathan Bryan

June 29, 2018

```
knitr::opts_chunk$set(fig.width=8, fig.height=5)
```

1. It was mentioned in the chapter that a cubic regression spline with one knot at ξ can be obtained using a basis of the form $x, x^2, x^3, (x-\xi)^3_+$, where $(x-\xi)^3_+ = (x-\xi)^3$ if $x > \xi$ and equals 0 otherwise. We will now show that a function of the form $f(x) = \beta 0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3_+$ is indeed a cubic regression spline, regardless of the values of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

(a) Find a cubic polynomial $f_1(x) = a_1 + b_1 x + c_1 x^2 + d_1 x^3$ such that $f(x) = f_1(x)$ for all $x \leq \xi$. Express a_1, b_1, c_1, d_1 in terms of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

$$
\begin{aligned}
f_1(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + (x-\xi)^3_+ \\
f_1(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3, \ \forall \ x \leq \xi \\
a_1 &= \beta_0, \ b_1 = \beta_1, \ c_1 = \beta_2, \ d_1 = \beta_3
\end{aligned}
$$

(b) Find a cubic polynomial $f_2(x) = a_2 + b_2 x + c_2 x^2 + d_2 x^3$ such that $f(x) = f_2(x)$ for all $x > \xi$. Express a_2, b_2, c_2, d_2 in terms of $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$. We have now established that $f(x)$ is a piecewise polynomial.

$$
\begin{aligned}
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3_+ \\
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)(x^2 - 2x\xi + \xi^2) \\
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 2x^2\xi + x\xi^2 - x^2\xi + 2x\xi^2 - \xi^3) \\
f_2(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x^3 - 3x^2\xi + 3x\xi^2 - \xi^3) \\
f_2(x) &= \beta_0 - \beta_4\xi^3 + (\beta_1 + 3\beta_4\xi^2)x + (\beta_2 - 3\beta_4\xi)x^2 + (\beta_3 + \beta_4)x^3 \\
a_2 &= \beta_0 - \beta_4\xi^3, \ b_2 = \beta_1 + 3\beta_4\xi^2, \ c_2 = \beta_2 - 3\beta_4\xi, \ d_1 = \beta_3 + \beta_4
\end{aligned}
$$

(c) Show that $f_1(\xi) = f_2(\xi)$. That is, $f(x)$ is continuous at ξ.

$$
f_1(\xi) = \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3
$$

$$
\begin{aligned}
f_2(x) &= \beta_0 - \beta_4\xi^3 + \beta_1\xi + 3\beta_4\xi^3 + \beta_2\xi^2 - 3\beta_4\xi^3 + \beta_3\xi^3 + \beta_4\xi^3 \\
f_2(x) &= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 + (\beta_4\xi^3 + 3\beta_4\xi^3 - 3\beta_4\xi^3 - \beta_4\xi^3) \\
f_2(x) &= \beta_0 + \beta_1\xi + \beta_2\xi^2 + \beta_3\xi^3 \\
f_1(\xi) &= f_2(\xi)
\end{aligned}
$$

(d) Show that $f_1'(\xi) = f_2'(\xi)$. That is, $f(x)$ is continuous at ξ.

$$f_1'(x) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2$$
$$f_1'(\xi) = \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2$$

$$f_2'(x) = \beta_1 + 3\beta_4 \xi^2 + 2\beta_2 x - 6\beta_4 \xi x + 3\beta_3 x^2 + 3\beta_4 x^2$$
$$f_2'(x) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2 + 3\beta_4 x^2 + 3\beta_4 \xi^2 - 6\beta_4 \xi x$$
$$f_2'(\xi) = \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2 + (3\beta_4 \xi^2 + 3\beta_4 \xi^2 - 6\beta_4 \xi^2)$$
$$f_2'(\xi) = \beta_1 + 2\beta_2 \xi + 3\beta_3 \xi^2$$
$$f_1'(\xi) = f_2'(\xi)$$

(e) Show that $f_1''(\xi) = f_2''(\xi)$. That is, $f''(x)$ is continuous at ξ. Therefore, $f(x)$ is indeed a cubic spline.

$$f_1''(x) = 2\beta_2 + 6\beta_3 x$$
$$f_1''(\xi) = 2\beta_2 + 6\beta_3 \xi$$

$$f_2''(x) = 2\beta_2 + 6\beta_3 x + 6\beta_4 x - 6\beta_4 \xi$$
$$f_2''(\xi) = 2\beta_2 + 6\beta_3 \xi + 6\beta_4 \xi - 6\beta_4 \xi$$
$$f_2''(\xi) = 2\beta_2 + 6\beta_3 \xi$$
$$f_1''(\xi) = f_2''(\xi)$$

2. Suppose that a curve $\hat{g}$ is computed to smoothly fit a set of n points using the following formula:

$$\hat{g} = arg\ \min_g (\Sigma_{i=1}^n (y_i - g(x_i))^2 + \lambda \int [g^{(m)}]^2\ dx)$$

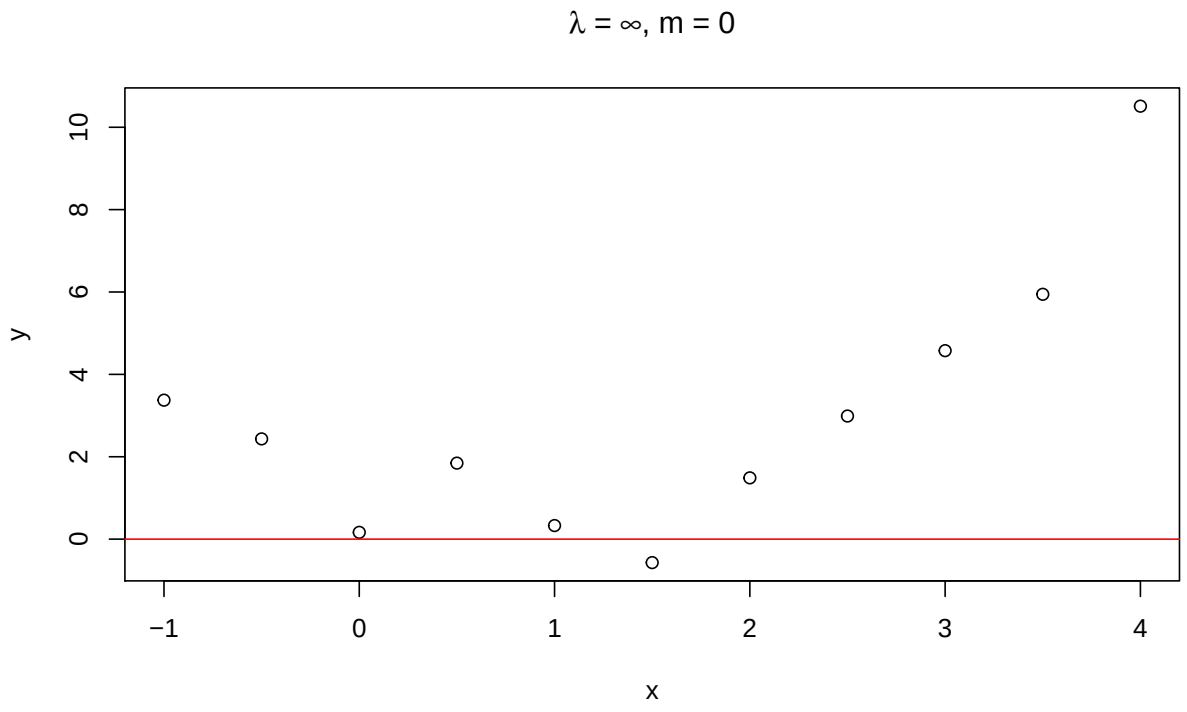
where $g^{(m)}$ represents the mth derivative of g (and $g^{(0)} = g$). Provide example sketches of $\hat{g}$ in each of the following scenarios.

(a) $\lambda = \infty, m = 0$.

Because this minimizes the values of the estimated curve itself, this forces $g^{(0)} = 0$

```
#create data
set.seed(1)
x = seq(-1,4,by=0.5)
e = rnorm(11,0,1)
y = (x-1)^2 + e
```

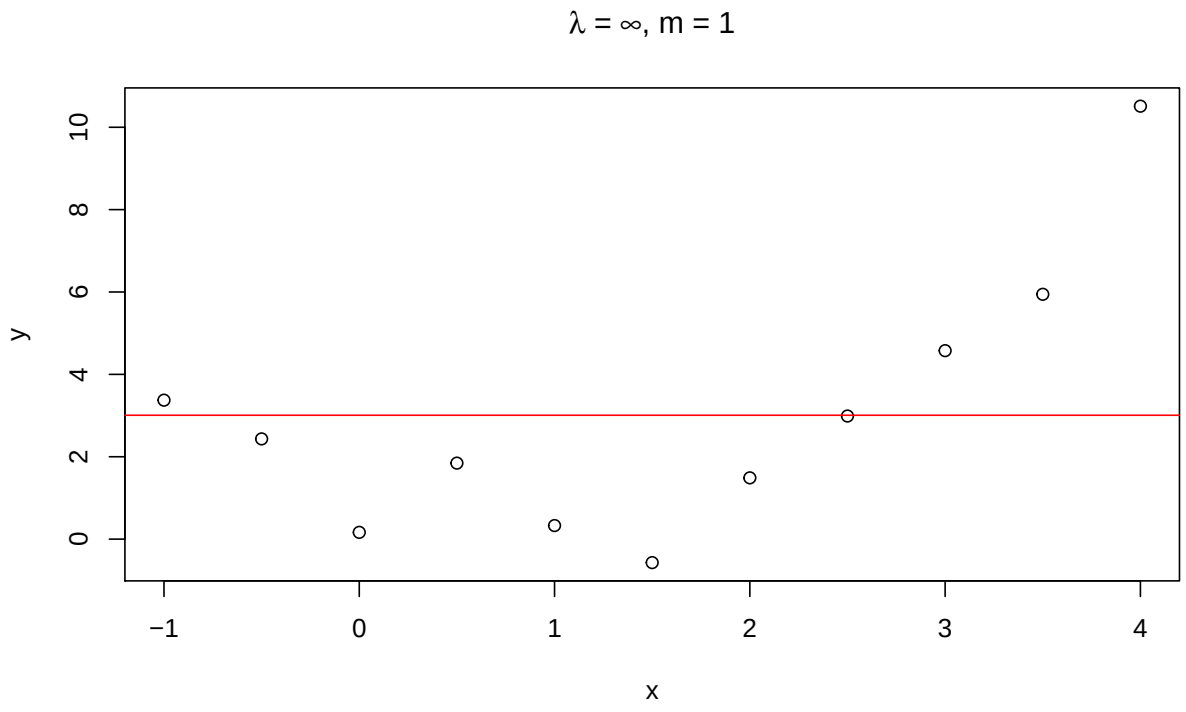
```
plot(x,y,
     main = expression(paste(lambda, " = ",infinity, ", m = 0")))
abline(h = 0, col="red")
```



(b) $\lambda = \infty, m = 1$**.**

This condition minimizes the the first derivative, therefore, the curve will be some constant, probably the midpoint of the response values to minimize the RSS as well.

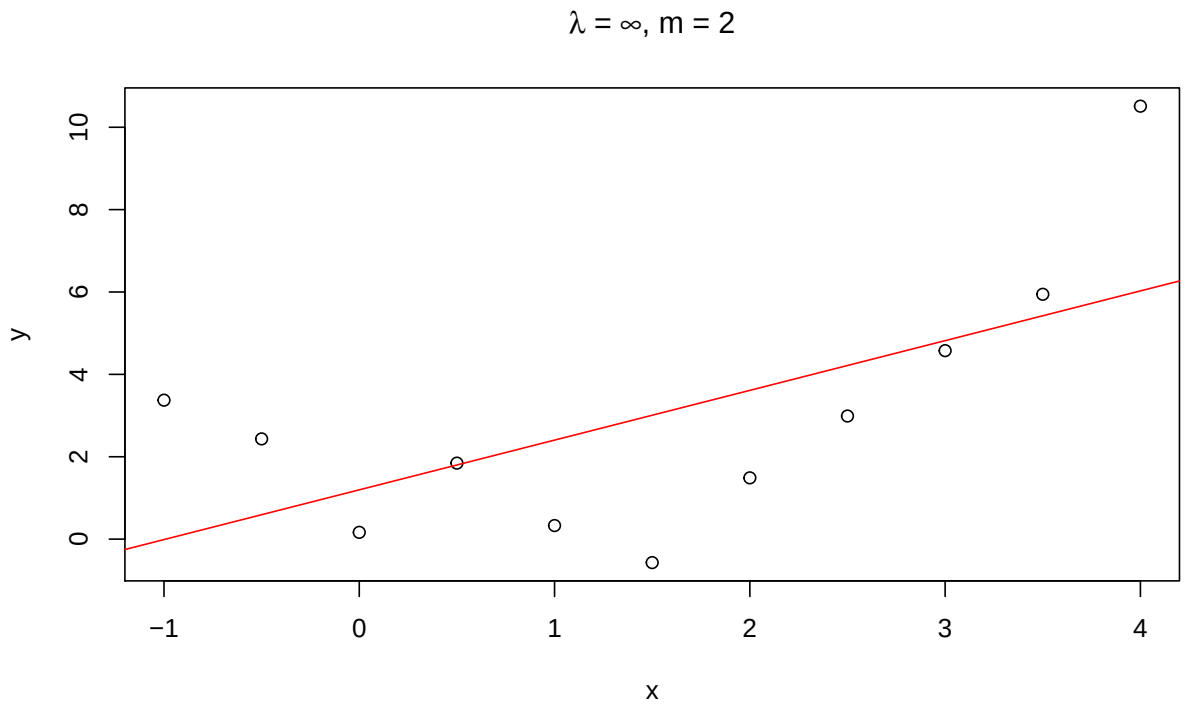
```
plot(x,y,
     main = expression(paste(lambda, " = ",infinity, ", m = 1")))
abline(h = mean(y), col="red")
```



(c) $\lambda = \infty, m = 2$**.**

This condition minimizes the second derivative inducing a linear trend on $g^{(0)}$.

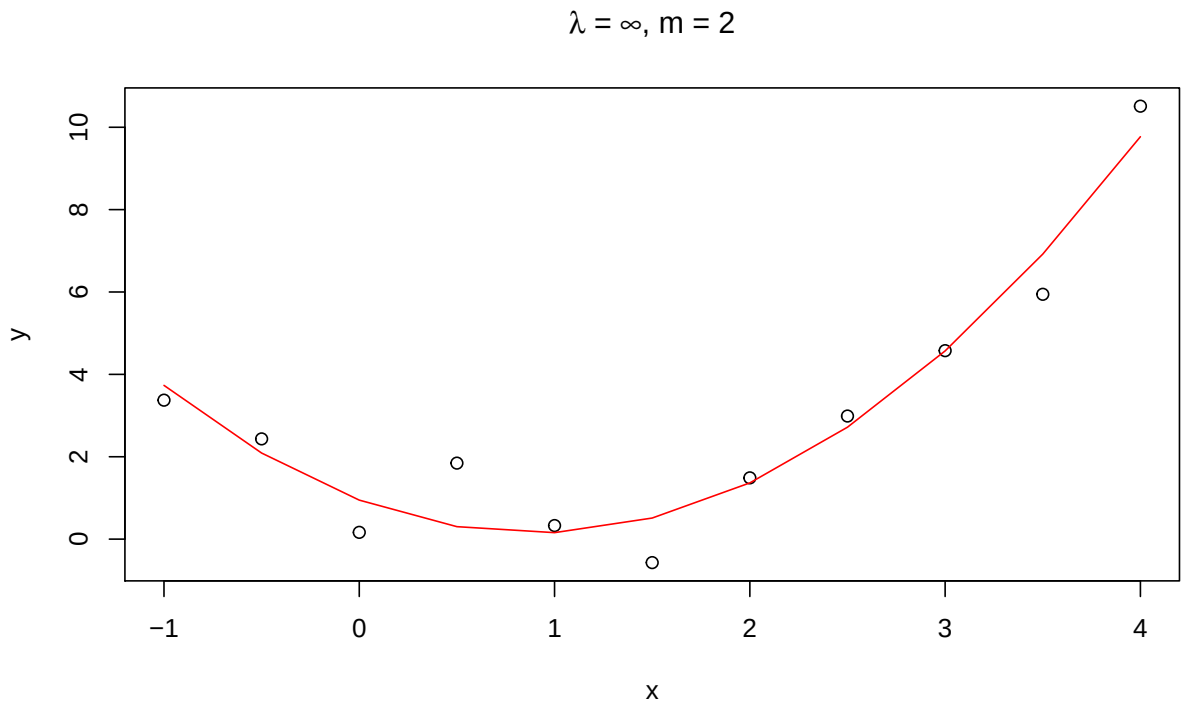
```
plot(x,y,
     main = expression(paste(lambda, " = ",infinity, ", m = 2")))
abline(lm(y ~ x), col="red")
```



(d) $\lambda = \infty, m = 3$**.**

This condition minimizes the third derivative inducing a quadratic trend on $g^{(0)}$.

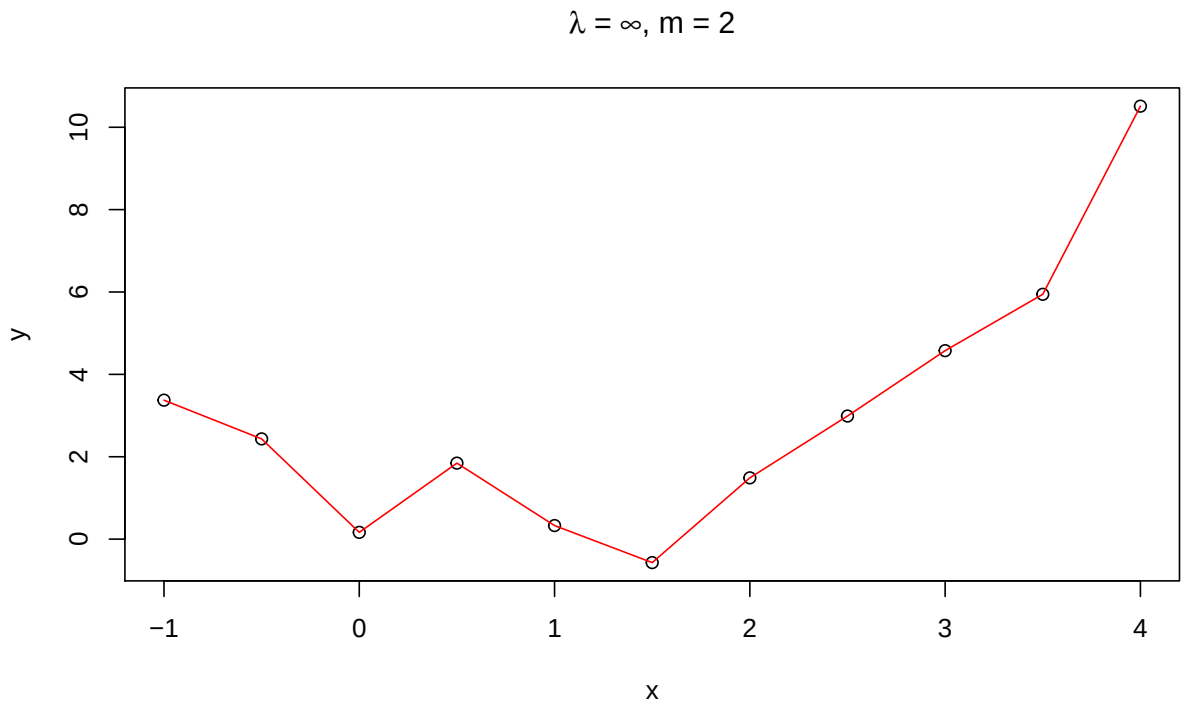
```
plot(x,y,
     main = expression(paste(lambda, " = ",infinity, ", m = 2")))
quad = lm(y ~ poly(x,2,raw=TRUE))
lines(x, predict(quad), col="red")
```



(e) $\lambda = 0, m = 3$.

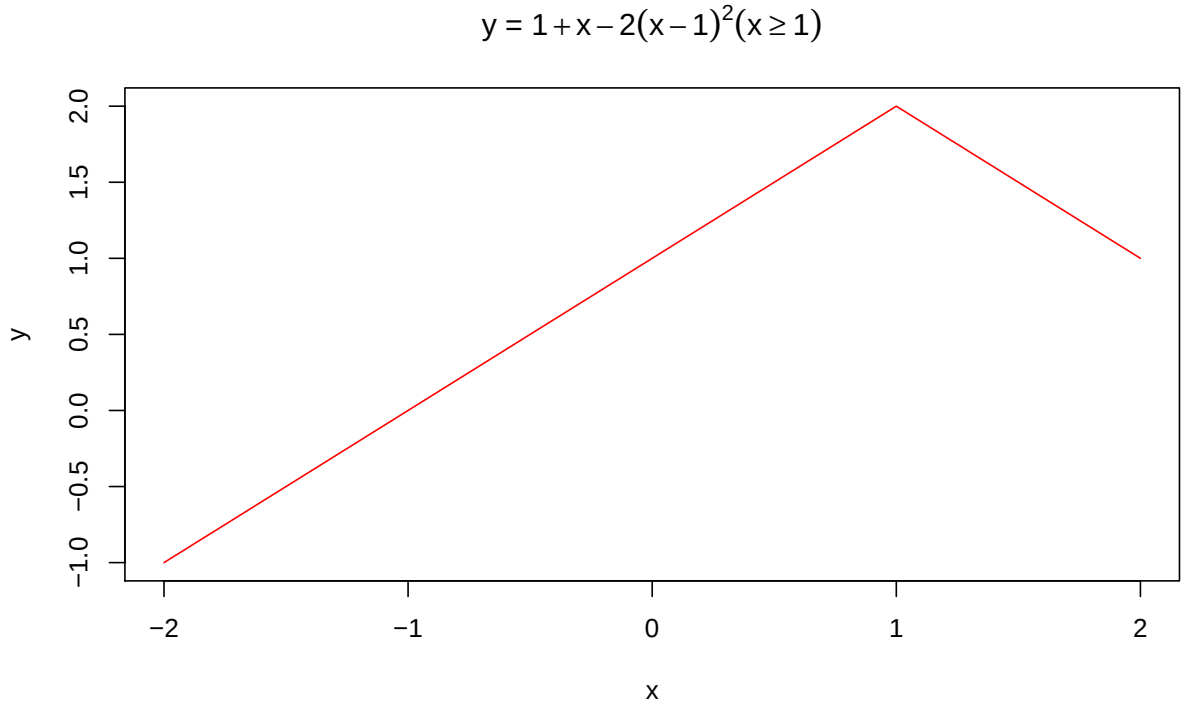
This condition recovers the interpoltaing spline for $g^{(0)}$.

```
plot(x,y,
     main = expression(paste(lambda, " = ",infinity, ", m = 2")))
spline = smooth.spline(x, y, df=11)
lines(predict(spline), col="red")
```



3. Suppose we fit a curve with basis functions $b_1(X) = X$, $b_2(X) = (X-1)^2\, I(X \geq 1)$. (Note that $I(X \geq 1)$ equals 1 for $X \geq 1$ 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.

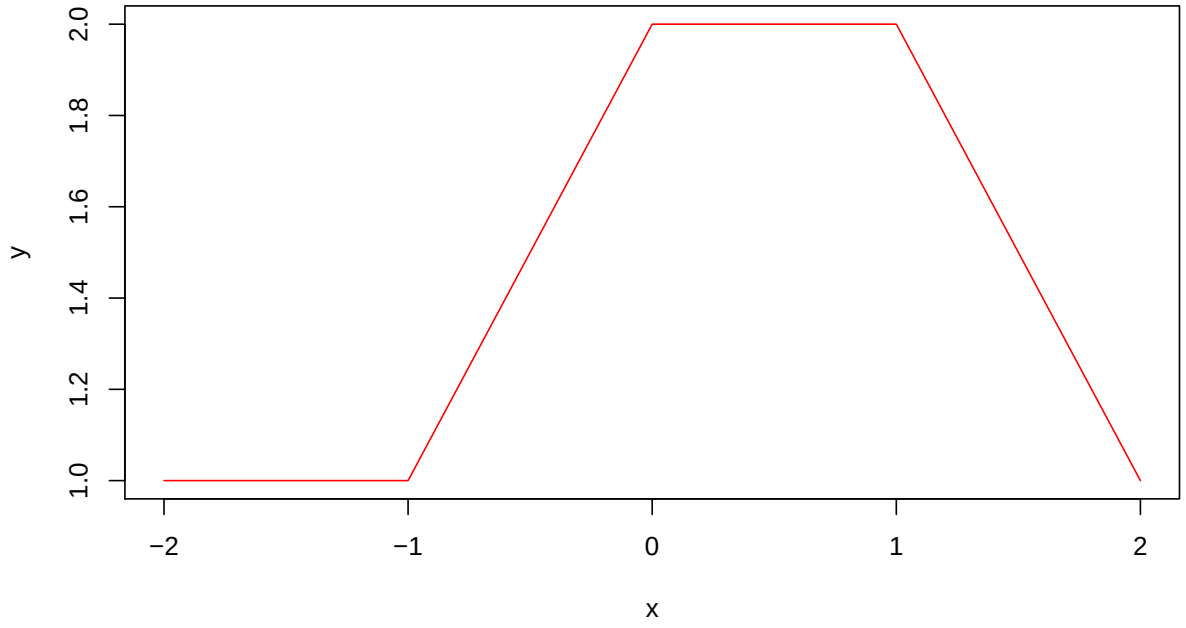
```
x = seq(-2,2,by=1)
y = 1 + x -2*(x-1)^2*(x >= 1)
plot(x,y,
     type="l",
     col="red",
     main = expression(paste(y," = ", 1 + x -2*(x-1)^2*(x >= 1))))
```



From $X = -2$ to $X = 1$ the intercept is 1 and slope is 1. When $X \geq 1$ the function becomes quadratic. The intecept is now 3, and the slope is now $4X - 3$.

4. Suppose we fit a curve with basis functions $b_1(X) = I(0 \leq X \leq 2) - (X-1)I(1 \leq X \leq 2)$, $b_2(X) = (X-3)I(3 \leq X \leq 4) + I(4 \leq X \leq 5)$. We fit the linear regression model $Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$,and obtain coefficient estimates $\hat{\beta}_0 = 1, \hat{\beta}_1 = 1, \hat{\beta}_2 = 3$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note the intercepts, slopes, and other relevant information.

```
x = seq(-2,2,by=1)
y = 1 + I((0 <= x) & (x <= 2)) - (x-1)*I((1 <= x) & (x <= 2)) +  3*((x-3)*I((3 <= x) & (x <= 4)) + I((4
plot(x,y,
     type="l",
     col="red"
)
```



From $X = -2$ to $X = -1$ the slope is 0 and the intercept is 1. From $X = -1$ to $X = 0$ the slope is 1 and the intercept is 2. From $X = 0$ to $X = 1$ the slope is 0 and the intercept is 2. From $X = 1$ to $X = 2$ the slope is -1 and the intercept is 3.

5. Consider two curves, $\hat{g_1}$ and $\hat{g_2}$, defined by

$$\hat{g_1} = arg\ \underset{g}{min}(\Sigma_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int [g^{(3)}]^2\ dx)$$

$$\hat{g_2} = arg\ \underset{g}{min}(\Sigma_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int [g^{(4)}]^2\ dx)$$

where $g^(m)$ represents the mth derivative of g.

(a) As $\lambda \to \infty$, will $\hat{g_1}$ or $\hat{g_2}$ have the smaller training RSS?

As λ approaches infinity and the degree of derivative increases in the penalized part of the optimization problem the flexibility of the model parameter's increases. Therefore, $\hat{g_2}$ will be more flexible and fit the training data better resulting in a lower training RSS.

(b) As $\lambda \to \infty$, will $\hat{g_1}$ or $\hat{g_2}$ have the smaller test RSS?

Because $\hat{g_1}$ is the less flexible model it will most likely have a lower variance and slightly higher bias resulting in a smaller test RSS.

(c) For $\lambda = 0$, will $\hat{g_1}$ or $\hat{g_2}$ have the smaller training and test RSS?

Both models will have the same training and test RSS because the penality is removed and the result is the same interpolating spline.

6. In this exercise, you will further analyze the Wage data set considered throughout this chapter.

(a) Perform polynomial regression to predict wage using age. Use cross-validation to select the optimal degree d for the polynomial. What degree was chosen, and how does this compare to the results of hypothesis testing using ANOVA? Make a plot of the resulting polynomial fit to the data.

```
library(ISLR)
set.seed(570)

rmse = function(preds, true){
  sqrt(sum((preds - true)^2)/length(true))
}

#initialize polynomial degrees, cv_rmse, and model lists
k = 5 #5-fold cross-validation
p = seq(1,10, by= 1)
mean.cv.rmse = rep(NA, length(p))
fit = rep(NA, length(p))

#fit polynomial models
for (i in 1:length(p)){
  cv.rmse = rep(NA, k)
  #5-fold CV
  for (j in 1:k){
    train = sample(1:nrow(Wage), nrow(Wage)*0.80, replace = FALSE)
    wage.train = Wage[train,]
    wage.test = Wage[-train,]
    preds = predict(lm(wage ~ poly(age,i, raw=TRUE),
                       data = wage.train),
                    newdata=wage.test)
    cv.rmse[j] = rmse(preds, wage.test$wage)
  }
  mean.cv.rmse[i] = mean(cv.rmse)
}

table1 = data.frame("Mean CV RMSE" = round(mean.cv.rmse,3))
rownames(table1) = seq(1,10,by=1)
knitr::kable(table1, caption = "Polynomial Regression Cross-validation")
```

Mean.CV.RMSE
39.483
42.015
39.568
39.864
39.207
39.896
39.904
39.672
40.616
39.465

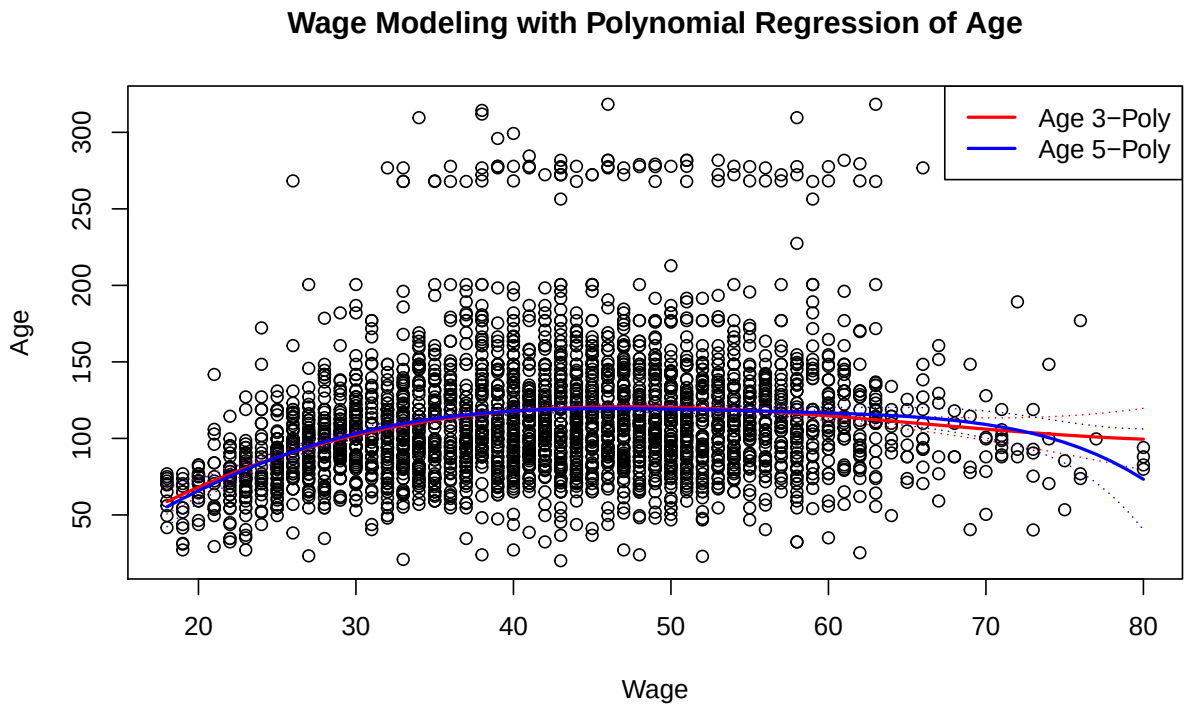
```r
#ANOVA
anova(lm(wage ~ poly(age,1, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,2, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,3, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,4, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,5, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,6, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,7, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,8, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,9, raw=TRUE), data = Wage),
      lm(wage ~ poly(age,10, raw=TRUE), data = Wage))
```

```
## Analysis of Variance Table
##
## Model  1: wage ~ poly(age, 1, raw = TRUE)
## Model  2: wage ~ poly(age, 2, raw = TRUE)
## Model  3: wage ~ poly(age, 3, raw = TRUE)
## Model  4: wage ~ poly(age, 4, raw = TRUE)
## Model  5: wage ~ poly(age, 5, raw = TRUE)
## Model  6: wage ~ poly(age, 6, raw = TRUE)
## Model  7: wage ~ poly(age, 7, raw = TRUE)
## Model  8: wage ~ poly(age, 8, raw = TRUE)
## Model  9: wage ~ poly(age, 9, raw = TRUE)
## Model 10: wage ~ poly(age, 10, raw = TRUE)
##    Res.Df     RSS Df Sum of Sq        F    Pr(>F)
## 1    2998 5022216
## 2    2997 4793430  1    228786 143.7638 < 2.2e-16 ***
## 3    2996 4777674  1     15756   9.9005  0.001669 **
## 4    2995 4771604  1      6070   3.8143  0.050909 .
## 5    2994 4770322  1      1283   0.8059  0.369398
## 6    2993 4766389  1      3932   2.4709  0.116074
## 7    2992 4763834  1      2555   1.6057  0.205199
## 8    2991 4763707  1       127   0.0796  0.777865
## 9    2990 4756703  1      7004   4.4014  0.035994 *
## 10   2989 4756701  1         3   0.0017  0.967529
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The model with the lowest cross-validated root mean square error using 5-fold cross-validation was a five-degree polynomial age model. ANOVA analysis suggests a cubic model explains most of the variance between wages and age. We plot both the cubic and five-degree polynomial models below.

```r
poly3 = lm(wage ~ poly(age,3, raw=TRUE), data = Wage)
poly5 = lm(wage ~ poly(age,5, raw=TRUE), data = Wage)
agelims = range(Wage$age)
age.grid = seq(from = agelims[1], to=agelims[2])

preds.poly3 = predict(poly3, newdata=list(age=age.grid), se=TRUE)
poly3.se.bands = cbind(preds.poly3$fit + 2*preds.poly3$se.fit, preds.poly3$fit - 2*preds.poly3$se)

preds.poly5 = predict(poly5, newdata=list(age=age.grid), se=TRUE)
poly5.se.bands = cbind(preds.poly5$fit + 2*preds.poly5$se.fit, preds.poly5$fit - 2*preds.poly5$se)

plot(Wage$age, Wage$wage,
     xlab = "Wage",
```

```
     ylab= "Age",
     main = "Wage Modeling with Polynomial Regression of Age")
lines(age.grid, preds.poly3$fit, lwd = 2, col = "red")
matlines(age.grid,poly3.se.bands, lwd=1,col="red", lty = 3)
lines(age.grid, preds.poly5$fit, lwd = 2, col = "blue")
matlines(age.grid,poly5.se.bands, lwd=1,col="blue", lty = 3)
legend("topright", legend = c("Age 3-Poly", "Age 5-Poly"),
       col = c("red", "blue"),
       lty = 1,
       lwd = 2
       )
```



(b) Fit a step function to predict wage using age, and perform cross-validation to choose the optimal number of cuts. Make a plot of the fit obtained.

```
set.seed(100)
#initialize cuts, cv_rmse, and model lists
k = 5 #5-fold cross-validation
p = seq(1,10, by= 1)
mean.cv.rmse = rep(NA, length(p) - 1)
fit = rep(NA, length(p))

#fit step function models
for (i in 2:length(p)){
  cv.rmse = rep(NA, k)
  #5-fold CV
  for (j in 1:k){
    train = sample(1:nrow(Wage), nrow(Wage)*0.80, replace = FALSE)
```

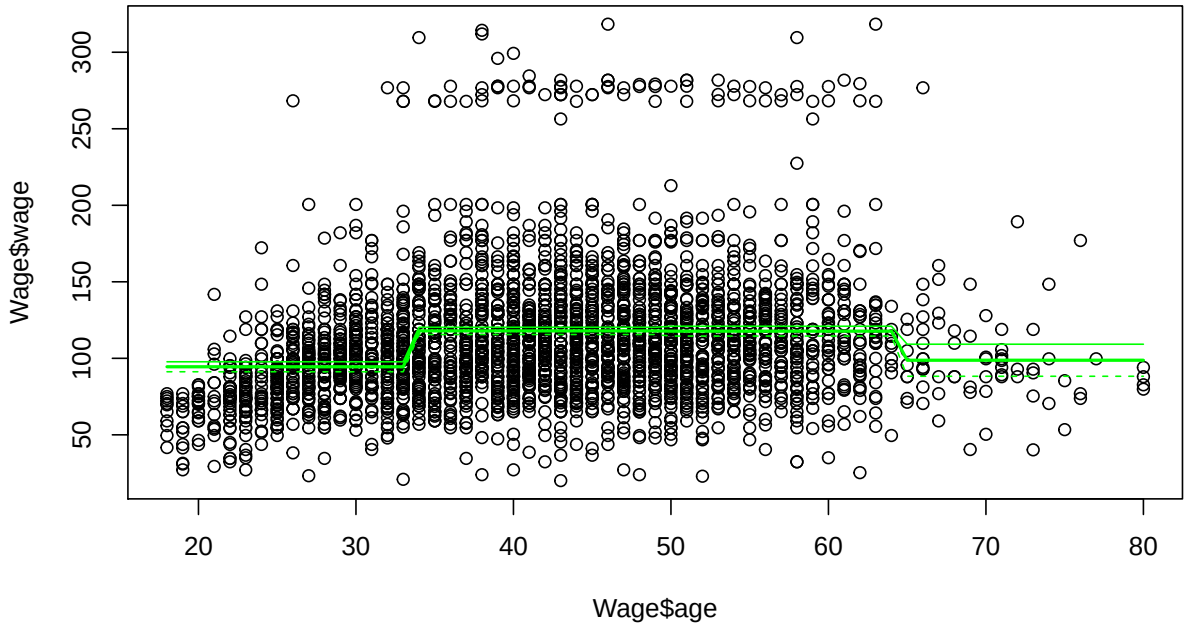
```
    wage.train = Wage[train,]
    wage.test = Wage[-train,]
    preds = predict(lm(wage ~ cut(Wage$age,i),
                       data = Wage,
                       subset = train),
                    newdata=cut(wage.test$age,i))
    cv.rmse[j] = rmse(preds, wage.test$wage)
  }
  mean.cv.rmse[i] = mean(cv.rmse)
}

mean.cv.rmse = mean.cv.rmse[-1]
table2 = data.frame("Mean CV RMSE" = round(mean.cv.rmse,3))
rownames(table2) = seq(2,10,by=1)
knitr::kable(table2, caption = "Step Function Regression Cross-validation")
```

Table 2: Step Function Regression Cross-validation

	Mean.CV.RMSE
2	91.990
3	95.124
4	91.822
5	93.823
6	95.355
7	99.979
8	97.628
9	97.836
10	94.762

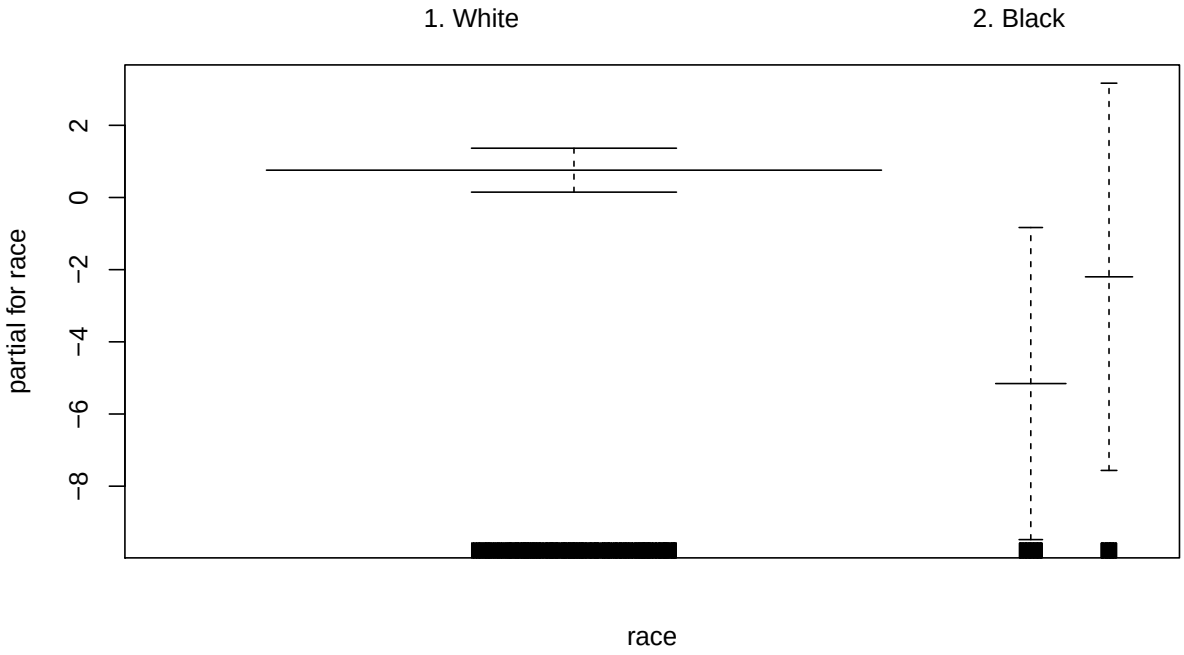
The 4-cut step function model of age has the lowest cross-validated root mean square error. A plot of the function is below.

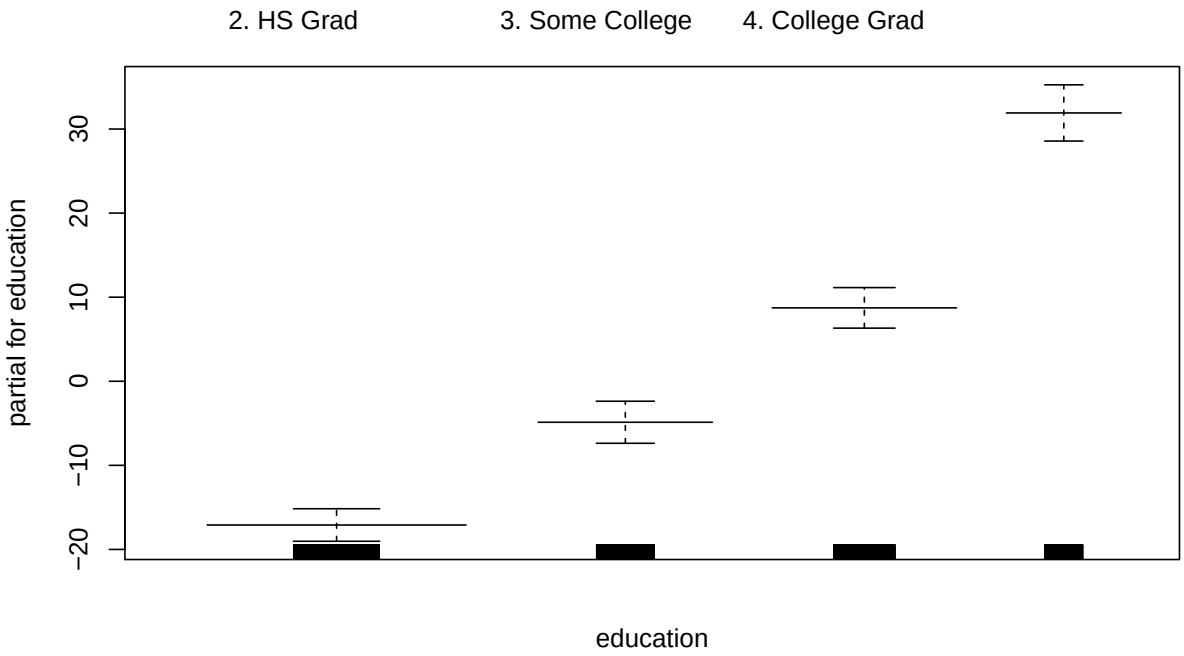
```
cut4 = lm(wage ~ cut(age,4), data = Wage, subset = train)

preds.cut4 = predict(cut4, newdata=list(age=age.grid), se=TRUE)
cut4.se.bands = cbind(preds.cut4$fit + 2*preds.cut4$se.fit, preds.cut4$fit - 2*preds.cut4$se)

plot(Wage$age, Wage$wage)
lines(age.grid, preds.cut4$fit, col ="green", lwd =2)
matlines(age.grid, cut4.se.bands, col ="green", lwd=1)
```

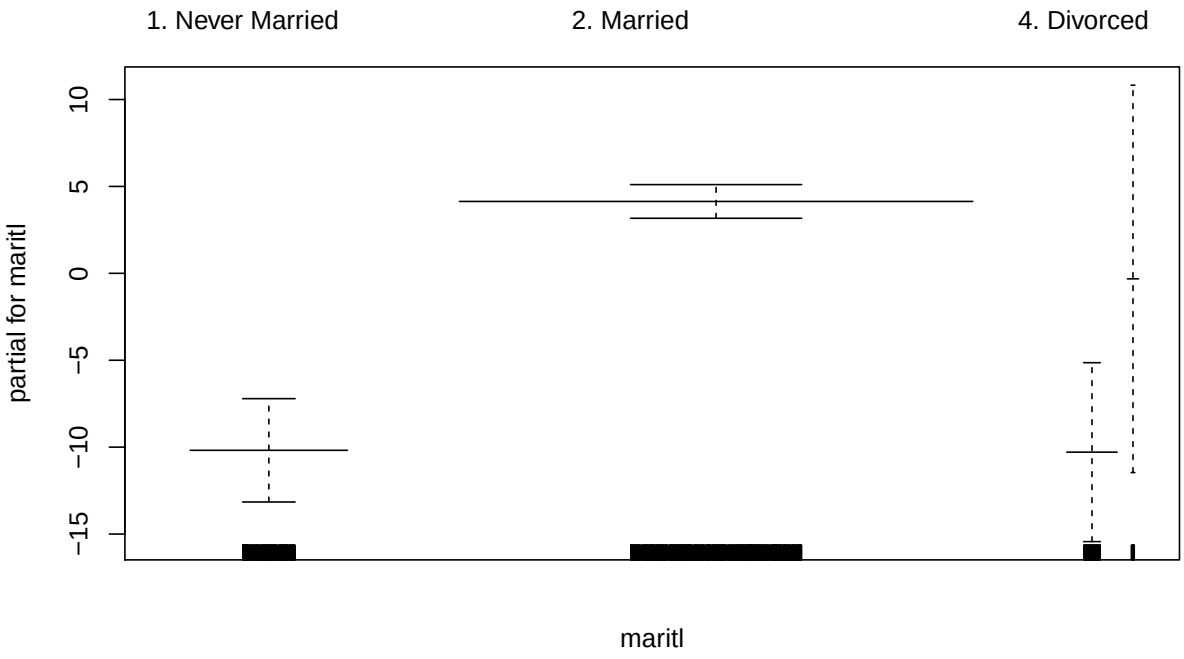


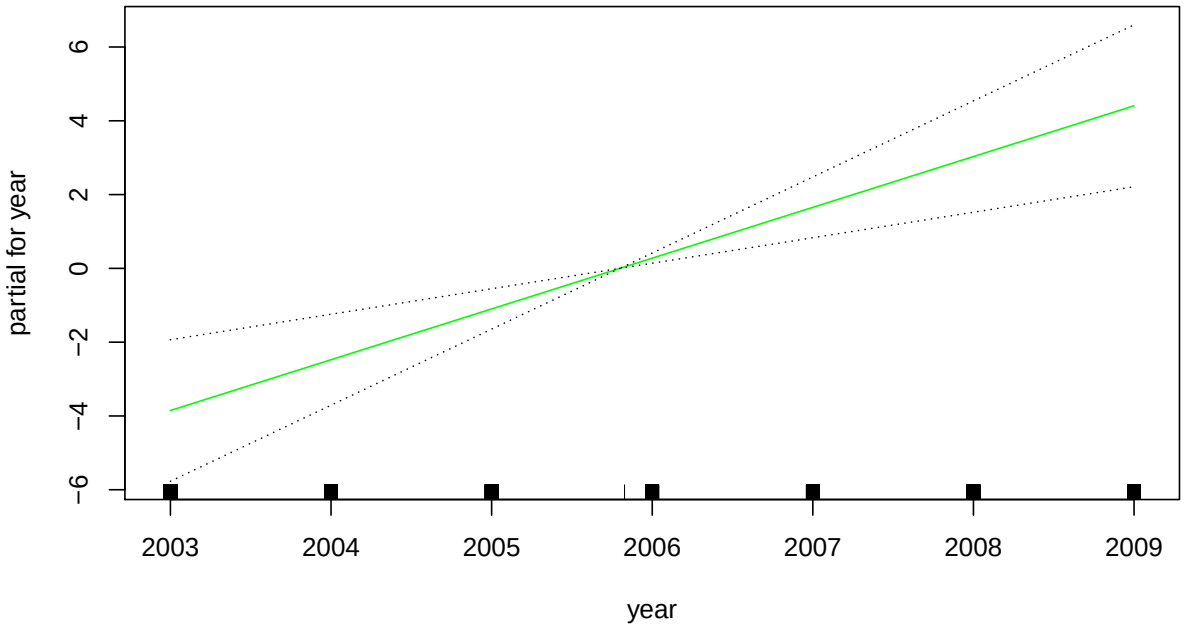
7. The Wage data set contains a number of other features not explored in this chapter, such as marital status (maritl), job class (jobclass), and others. Explore the relationships between some of these other predictors and wage, and use non-linear fitting techniques in order to fit flexible models to the data. Create plots of the results obtained, and write a summary of your findings.

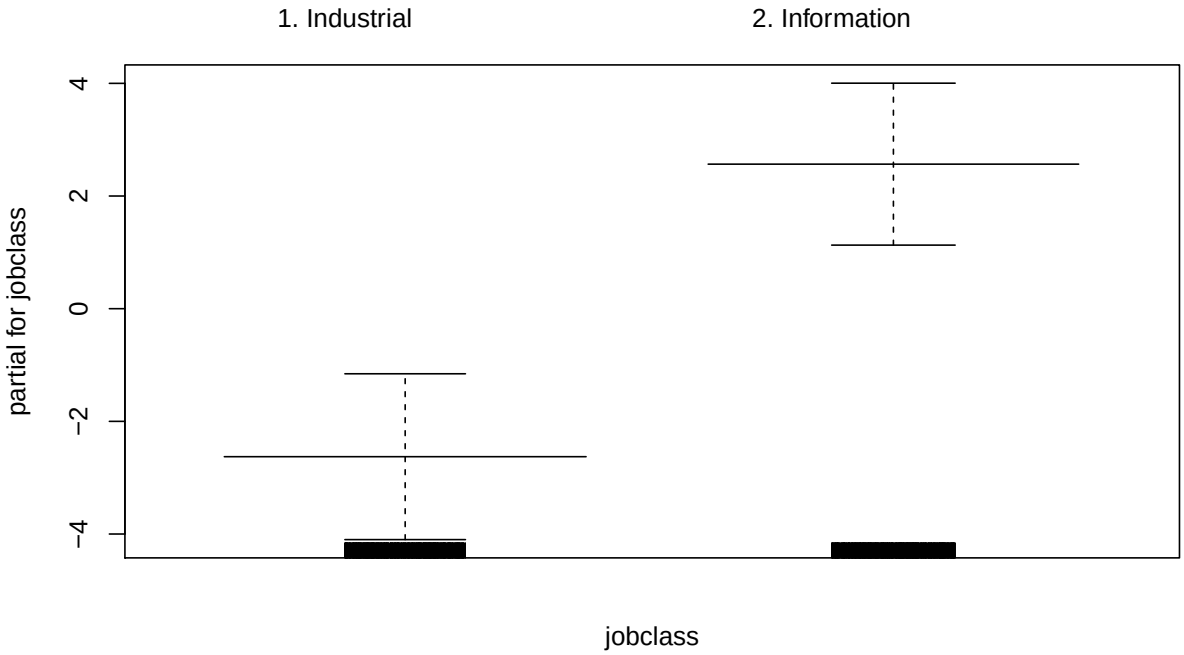
```
library(gam)
gam1 = gam(wage ~ race + education + maritl + year + jobclass + s(age,4),
           data = Wage,
           subset= (education!="1. < HS Grad") & (race!="4. Other") & (maritl!="3. Widowed"))
plot(gam1, se=TRUE, col="green")
```

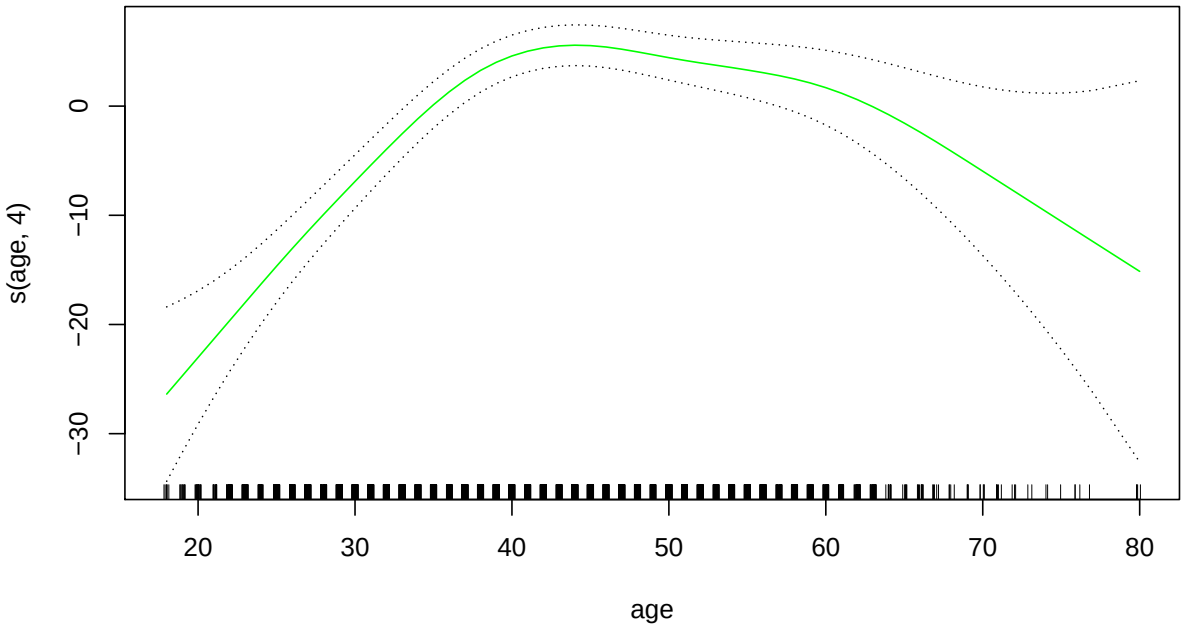

1. White
2. Black
partial for race
2
0
−2
−4
−6
−8
race


2. HS Grad
3. Some College
4. College Grad
partial for education
30
20
10
0
−10
−20
education


1. Never Married
2. Married
4. Divorced
partial for maritl
10
5
0
−5
−10
−15
maritl


partial for year
6
4
2
0
−2
−4
−6
2003
2004
2005
2006
2007
2008
2009
year






```
summary(gam1)
```

```
##
## Call: gam(formula = wage ~ race + education + maritl + year + jobclass +
```

```
##     s(age, 4), data = Wage, subset = (education != "1. < HS Grad") &
##     (race != "4. Other") & (maritl != "3. Widowed"))
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -109.440  -20.394   -2.613   13.873  212.155
##
## (Dispersion Parameter for gaussian family taken to be 1281.68)
##
##     Null Deviance: 4828594 on 2689 degrees of freedom
## Residual Deviance: 3428494 on 2675 degrees of freedom
## AIC: 26900.29
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## race         2   46140   23070  18.000 1.717e-08 ***
## education    3  912518  304173 237.323 < 2.2e-16 ***
## maritl       3  170545   56848  44.355 < 2.2e-16 ***
## year         1   22833   22833  17.815 2.515e-05 ***
## jobclass     1   19617   19617  15.306 9.371e-05 ***
## s(age, 4)    1   42178   42178  32.909 1.074e-08 ***
## Residuals 2675 3428494    1282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df Npar F     Pr(F)
## (Intercept)
## race
## education
## maritl
## year
## jobclass
## s(age, 4)         3  21.38 1.104e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using a generalized additive model we model year linearly, age with a 4-knot smoothing spline, and jobclass, race, education, maritl as categorical variables. We remove those without a high school education, those who identified as race "other", and widowed individuals because the initial confidence intervals for the parameter estimates were too wide. We find that all of the covariates are significant at an $\alpha = 0.05$ level. White race, increasing education, being married, advancing year, information-related job class, and middle age are all associated with increasing wage values.

8. Fit some of the non-linear models investigated in this chapter to the Auto data set. Is there evidence for non-linear relationships in this data set? Create some informative plots to justify your answer.

```
auto.poly = lm(mpg ~ year + displacement + poly(horsepower,2, raw = TRUE) + weight + acceleration,
               data = Auto)
auto.lr1 = gam(mpg ~ year + displacement + lo(horsepower,span = 0.2) + weight + acceleration,
               data = Auto)
```

```
auto.lr2 = gam(mpg ~ year + lo(horsepower,span = 0.2) + weight + acceleration,
               data = Auto)
auto.ss1 = gam(mpg ~ year + s(horsepower,df = 4) + weight + acceleration,
               data = Auto)
auto.ss2 = gam(mpg ~ year + displacement + s(horsepower,df = 4) + weight + acceleration,
               data = Auto)
```

We fit a linear regression with a quadratic polynomial term for horsepower after testing for polynomial forms of the other numerical covariates. Horsepower was determined to be the only covariate likely to have a polynomial form after significance testing for each numerical covariate with a third-degree polynomial form at the $\alpha = 0.05$ level. Local regression (span $= 0.2$) and smoothing splines (df $= 4$) were then used to determine the best non-linear functional form to model horsepower. Using the quadratic polynomial form for horsepower we find that year, horsepower, weight, and acceleration are all significantly associated with wage. Displacement was not significant at the $\alpha = 0.05$ level, however, it was using the other non-linear methods for horsepower.

```
anova(auto.ss1,auto.ss2)
```

```
## Analysis of Deviance Table
##
## Model 1: mpg ~ year + s(horsepower, df = 4) + weight + acceleration
## Model 2: mpg ~ year + displacement + s(horsepower, df = 4) + weight +
##     acceleration
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       384     3458.6
## 2       383     3403.1  1   55.491  0.01245 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(auto.lr1,auto.lr2)
```

```
## Analysis of Deviance Table
##
## Model 1: mpg ~ year + displacement + lo(horsepower, span = 0.2) + weight +
##     acceleration
## Model 2: mpg ~ year + lo(horsepower, span = 0.2) + weight + acceleration
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1    376.97     3350.6
## 2    377.97     3406.2 -1  -55.587  0.01239 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using nested models, where displacement is removed, for both the local regression and smoothing spline we find that displacement is a signiciant predictor (p-vale $= 0.01$) and should be included in the model.

```
set.seed(650)
cv.rmse.auto = matrix(rep(NA,15), nrow = 3, ncol = 5)
k = 5
for (i in 1:k){
  train = sample(1:nrow(Auto), nrow(Auto)*.80, replace = FALSE)
  auto.train = Auto[train,]
  auto.test = Auto[-train,]
  auto.poly = lm(mpg ~ year + displacement + poly(horsepower,2, raw = TRUE) + weight + acceleration,
                 data = auto.train)
  auto.lr = gam(mpg ~ year + displacement + lo(horsepower,span = 0.2) + weight + acceleration,
                data = auto.train)
```

```
  auto.ss = gam(mpg ~ year + s(horsepower,df = 4) + weight + acceleration,
                data = auto.train)
  auto.poly.preds = predict(auto.poly, newdata = auto.test)
  auto.lr.preds = predict(auto.lr, newdata = auto.test)
  auto.ss.preds = predict(auto.ss, newdata = auto.test)
  cv.rmse.auto[,i] = c(rmse(auto.poly.preds,auto.test$mpg),
                       rmse(auto.lr.preds,auto.test$mpg),
                       rmse(auto.ss.preds,auto.test$mpg))
}
table3 = round(rowMeans(cv.rmse.auto),3)
names(table3) = c("HP 2-Poly", "HP Local Regression", "HP Smoothing Spline")
knitr::kable(table3, caption= "5-Fold CV RMSE")
```

Table 3: 5-Fold CV RMSE

HP 2-Poly	3.132
HP Local Regression	3.111
HP Smoothing Spline	3.057

Using 5-fold cross-validation we find that modeling horsepower with a smoothing spline (df = 4) gives the lowest cross-validated root mean square error, suggesting this is the best model.

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.

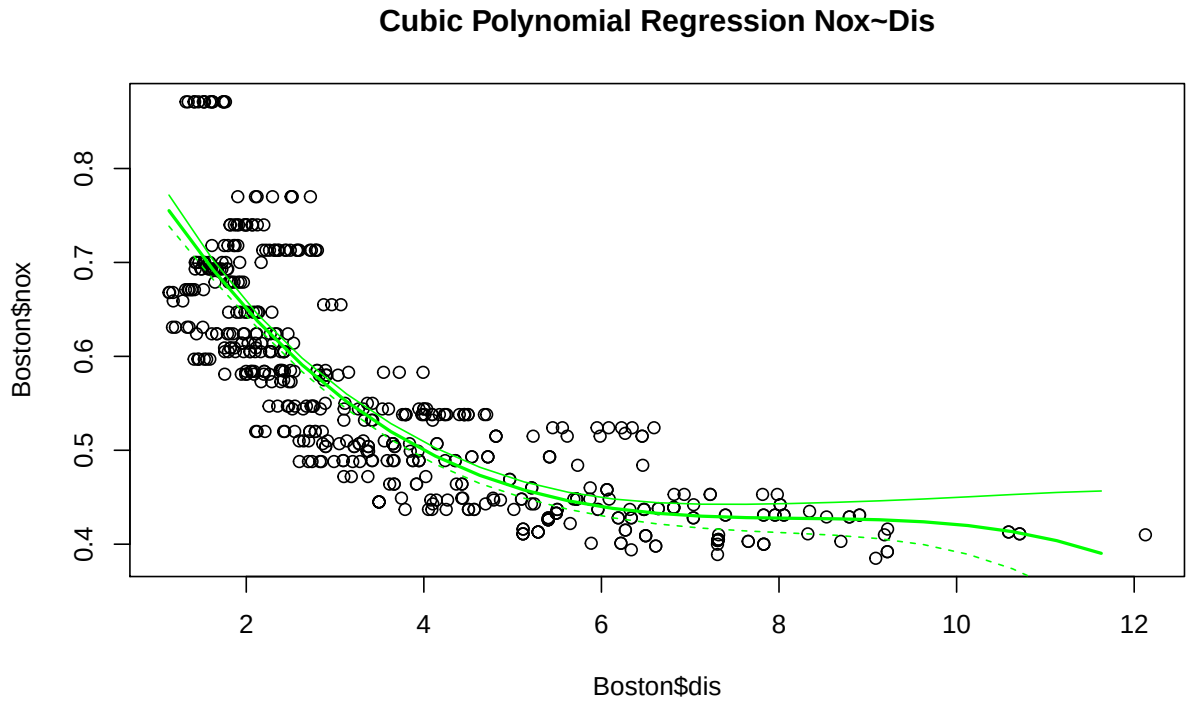
(a) Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

```
library(MASS)

#model
boston.poly = lm(nox ~ poly(dis,3,raw = TRUE),
                 data = Boston)

#summary
summary(boston.poly)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3, raw = TRUE), 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)                 0.9341281  0.0207076  45.110  < 2e-16 ***
## poly(dis, 3, raw = TRUE)1 -0.1820817  0.0146973 -12.389  < 2e-16 ***
## poly(dis, 3, raw = TRUE)2  0.0219277  0.0029329   7.476 3.43e-13 ***
## poly(dis, 3, raw = TRUE)3 -0.0008850  0.0001727  -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
#plot
dis.grid = seq(from = range(Boston$dis)[1],
               to = range(Boston$dis)[2], by = 0.5)
preds.boston.poly = predict(boston.poly, newdata = list(dis = dis.grid), se = TRUE)
se.bands = cbind(preds.boston.poly$fit + 2*preds.boston.poly$se.fit,
                 preds.boston.poly$fit - 2*preds.boston.poly$se)


plot(Boston$dis,Boston$nox,
     main = "Cubic Polynomial Regression Nox~Dis")

lines(dis.grid,preds.boston.poly$fit,
      col = "green",
      lwd = 2)

matlines(dis.grid,se.bands,
      col = "green",
      lwd = 1)
```



(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.

```
poly.fits = seq(1,10, by = 1)
boston.RSS = rep(NA, 10)
for (i in poly.fits){
  boston.poly = lm(nox ~ poly(dis, i),
                   data = Boston)
  boston.RSS[i] = sum(summary(boston.poly)$residuals^2)
}
names(boston.RSS) = poly.fits
knitr::kable(round(boston.RSS,3), caption = "RSS for Polynomial Fits of Nox~Dis")
```

2.769
2.035
1.934
1.933
1.915
1.878
1.849
1.836
1.833
1.832

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

```
set.seed(354)
k = 10
boston.cv.RMSE = matrix(NA, nrow = 10, ncol = k)
for (i in poly.fits){
  for (j in 1:k){
    train = sample(1:nrow(Boston),
                   size = nrow(Boston)*(k-1)/k,
                   replace = FALSE)
    boston.train = Boston[train,]
    boston.test = Boston[-train,]
    boston.poly = lm(nox ~ poly(dis,i),
                     data = boston.train)
    boston.preds = predict(boston.poly,
                           newdata = boston.test)
    boston.cv.RMSE[i,j] = rmse(boston.preds, boston.test$nox)
  }
}
table4 = round(rowMeans(boston.cv.RMSE),3)
names(table4) = poly.fits
knitr::kable(round(table4,3), caption = "RMSE of Polynomial Regression Nox~Dis")
```

0.072
0.068
0.063
0.061
0.064
0.060
0.064

0.061
0.062
0.062

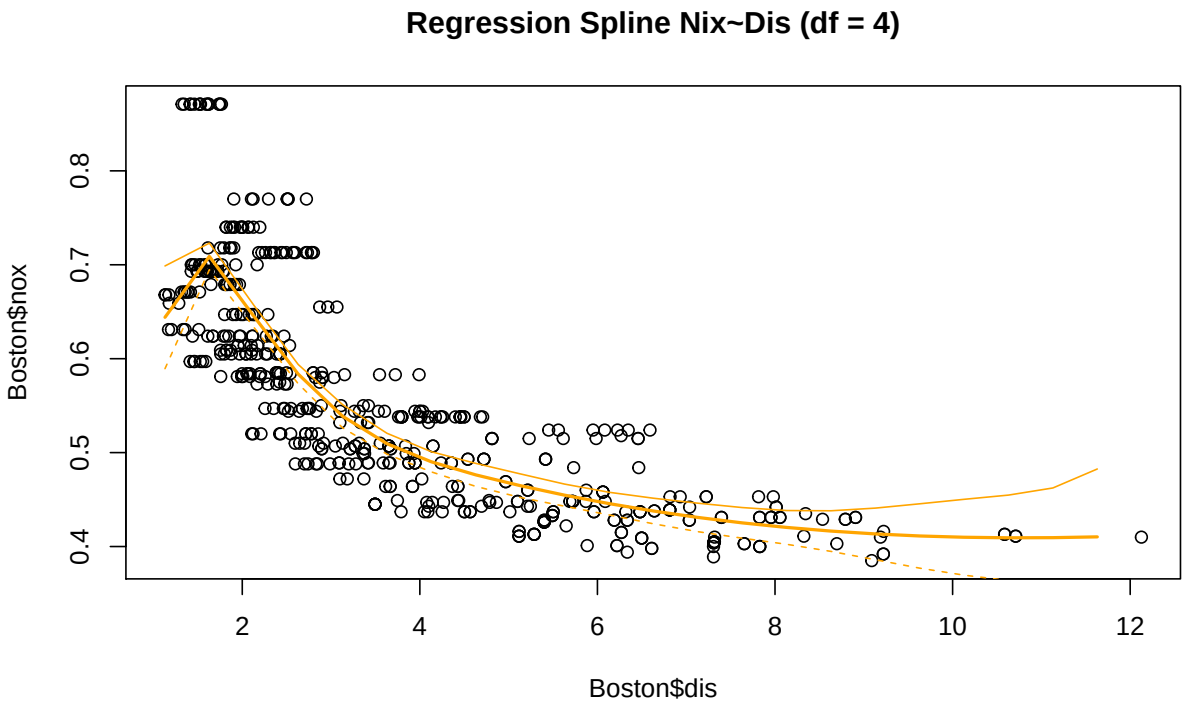
Using 10-fold cross-validation, the 6-degree polynomial model had the lowest cross-validated root mean square error (0.60) and is therefore the best model.

(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.

```
set.seed(900)
#knot selection
knots = as.numeric(quantile(Boston$dis))

#model
boston.bs = lm(nox~bs(dis,
                      knots = knots,
                      degree = 4),
               data = Boston)

#prediction for plotting
boston.bs.preds = predict(boston.bs,
                          newdata = list(dis = dis.grid),
                          se = TRUE)
se.bands = cbind(boston.bs.preds$fit + 2*boston.bs.preds$se.fit,
                 boston.bs.preds$fit - 2*boston.bs.preds$se)
#summary
summary(boston.bs.preds)
```

```
##                Length Class  Mode
## fit            22     -none- numeric
## se.fit         22     -none- numeric
## df              1     -none- numeric
## residual.scale  1     -none- numeric
```

```
#plots
plot(Boston$dis, Boston$nox,
     main = "Regression Spline Nix~Dis (df = 4)")
lines(dis.grid, boston.bs.preds$fit,
      lwd = 2,
      col = "orange")
matlines(dis.grid, se.bands, lwd = 1, col = "orange")
```



Knots were were selected at the 0, 25, 50, 75, and 100 percentiles of the dis data.

(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.

```
df.fits = seq(1,10, by = 1)
boston.RSS2 = rep(NA, length(df.fits))
for (i in df.fits){
  boston.bs = lm(nox ~ bs(dis,
                          knots = knots,
                          df = i),
                 data = Boston)
  boston.RSS2[i] = sum(boston.bs$residuals^2)
}
names(boston.RSS2) = df.fits
knitr::kable(round(boston.RSS2,3), caption = "RSS for Regression Spline Fits of Nox~Dis")
```

1.834
1.834
1.834
1.834
1.834
1.834
1.834
1.834
1.834
1.834

Evaluting degrees of freedom from 1 to 10 and the same knots from part (d) we see that the RSS is the same for each level of df.

(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.

```
set.seed(222)
k = 10
boston.cv.RMSE = matrix(NA, nrow = 10, ncol = k)
for (i in df.fits){
  for (j in 1:k){
    train = sample(1:nrow(Boston),
                   size = nrow(Boston)*(k-1)/k,
                   replace = FALSE)
    boston.train = Boston[train,]
    boston.test = Boston[-train,]
    boston.bs = lm(nox ~ bs(dis,
                            knots = knots,
                            Boundary.knots = c(range(Boston$dis)[1],
                                               range(Boston$dis)[2]),
                            df = i),
                   data = boston.train)
    boston.preds = predict(boston.bs,
                           newdata = boston.test)
    boston.cv.RMSE[i,j] = rmse(boston.preds, boston.test$nox)
  }
}
table4 = round(rowMeans(boston.cv.RMSE),3)
names(table4) = df.fits
knitr::kable(table4, caption = "RMSE of Regression Spline Nox~Dis")
```

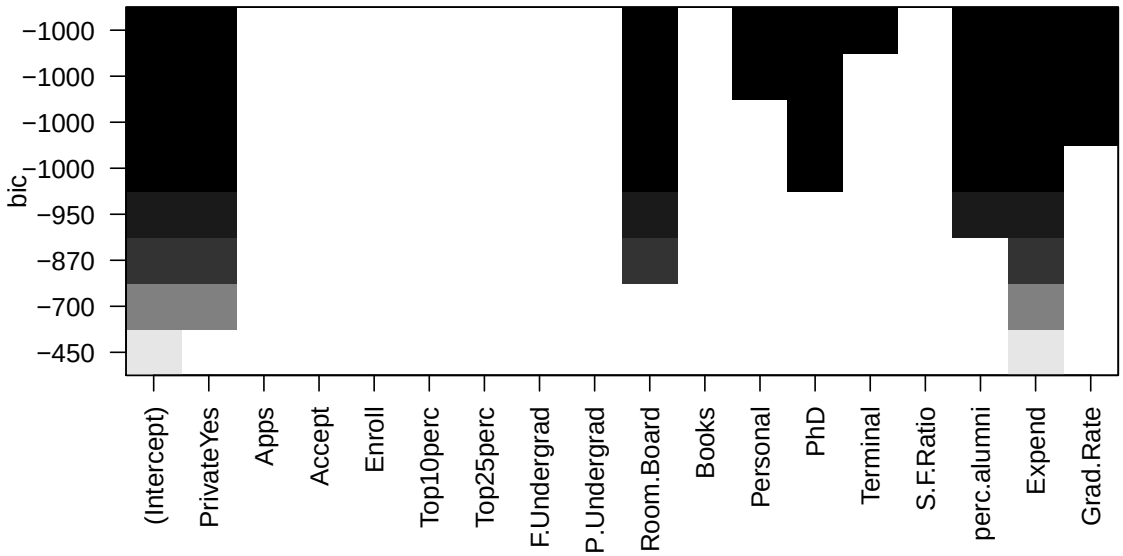
0.062
0.061
0.059
0.064
0.061
0.064
0.062
0.057
0.059
0.059

Using 10-fold cross validation, the model with eight degress of freedom had the lowest cross-validated root mean square error and is therefore the best for the regression spline.

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 stepwise selection on the training set in order to identify a satisfactory model that uses just a subset of the predictors.

```
library(leaps)

#validation set
train = sample(1:nrow(College),
               size = nrow(College)*.75)
college.train = College[train,]
college.test = College[-train,]

#model
college.fwd = regsubsets(Outstate ~.,
                         data= College,
                         method = "forward")

#summary
plot(college.fwd)
```




```
which.min(summary(college.fwd)$bic)
```

```
## [1] 8
```

```
coef(college.fwd, which.min(summary(college.fwd)$bic))
```

```
##   (Intercept)    PrivateYes    Room.Board      Personal           PhD
## -3427.0217976  2751.0006770     0.9097872    -0.3216904    18.8340453
##      Terminal   perc.alumni        Expend     Grad.Rate
##    24.7935526    43.3961323     0.2220780    28.3705072
```

Forward selection included 8 covariates with the coefficients listed above.

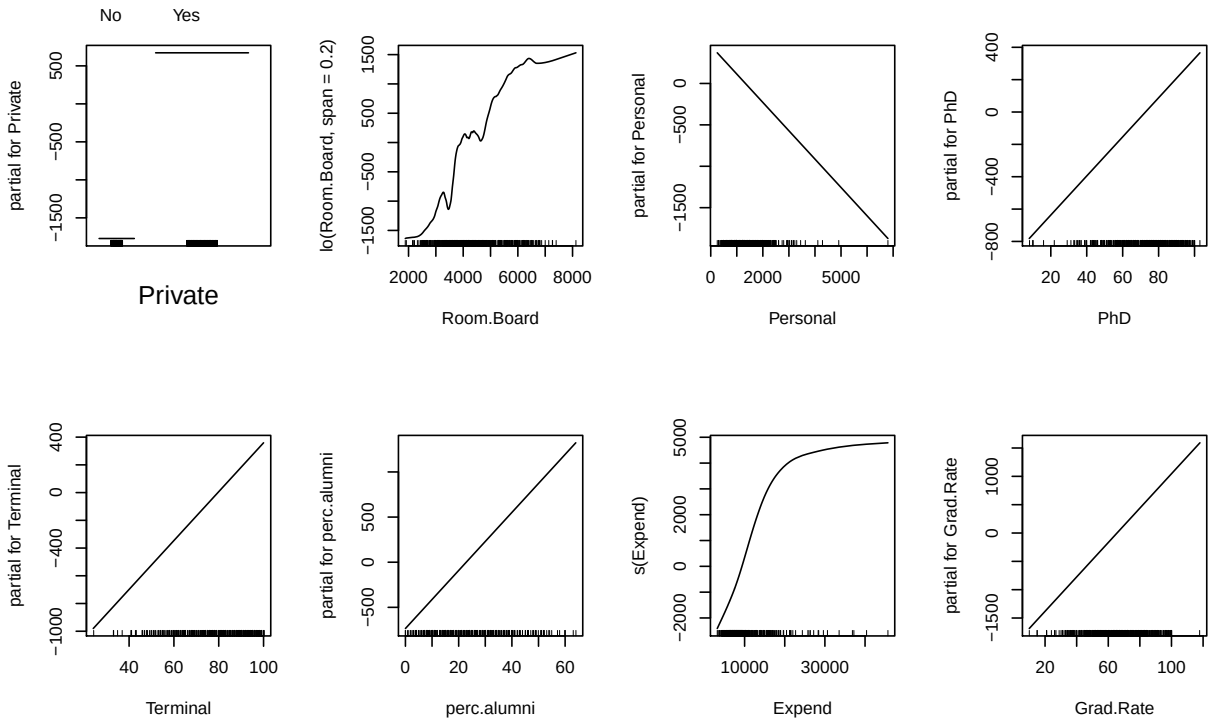
(b) Fit a GAM on the training data, using out-of-state tuition as the response and the features selected in the previous step as the predictors. Plot the results, and explain your findings.

```
college.gam = gam(Outstate ~ Private +
                             lo(Room.Board, span=0.2) +
                             Personal +
                             PhD +
                             Terminal +
                             perc.alumni +
                             s(Expend) +
                             Grad.Rate, data = college.train
                  )
summary(college.gam)
```

```
##
## Call: gam(formula = Outstate ~ Private + lo(Room.Board, span = 0.2) +
##     Personal + PhD + Terminal + perc.alumni + s(Expend) + Grad.Rate,
##     data = college.train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7176.90 -1078.28    55.93  1260.75  4527.07
##
## (Dispersion Parameter for gaussian family taken to be 3424098)
##
##     Null Deviance: 9641557028 on 581 degrees of freedom
## Residual Deviance: 1920338078 on 560.8304 degrees of freedom
## AIC: 10431.39
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                                Df     Sum Sq    Mean Sq  F value    Pr(>F)
## Private                      1.00 2662750808 2662750808 777.6503 < 2.2e-16
## lo(Room.Board, span = 0.2)   1.00 2364817964 2364817964 690.6397 < 2.2e-16
## Personal                     1.00   31014619   31014619   9.0577 0.0027334
## PhD                          1.00  650379513  650379513 189.9419 < 2.2e-16
## Terminal                     1.00   44816693   44816693  13.0886 0.0003239
## perc.alumni                  1.00  334727461  334727461  97.7564 < 2.2e-16
## s(Expend)                    1.00  771420784  771420784 225.2917 < 2.2e-16
## Grad.Rate                    1.00   97821420   97821420  28.5685 1.318e-07
## Residuals                  560.83 1920338078    3424098
##
## Private                      ***
## lo(Room.Board, span = 0.2) ***
## Personal                     **
## PhD                          ***
## Terminal                     ***
## perc.alumni                  ***
## s(Expend)                    ***
## Grad.Rate                    ***
## Residuals
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
```

```
##                               Npar Df  Npar F     Pr(F)
## (Intercept)
## Private
## lo(Room.Board, span = 0.2)     9.2  1.7831   0.06713 .
## Personal
## PhD
## Terminal
## perc.alumni
## s(Expend)                      3.0 22.9442 5.085e-14 ***
## Grad.Rate
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
par(mfrow=c(2,4))
plot(college.gam)
```



All 8 covariates were found to be significant. Each covariate was first modelled as a smoothing spline to test for non-linearity between the covariate and the response. Only rooom and board costs, as well as, instructional expenditure per student were found to have a signiciant non-linear relation to the response. Room and board were modelled with local regression of span equal to 20 percent and the instructional spending per student was modelled as a smoothing spline with df = 3. For the linear terms we observe that increasing percent of faculty with terminal degree, percent of faculty with Ph.D's, percent of alumni who donate, and graduation rate are all associated with greater out-of-state tuition. Only estimated personal spending was inversely associated with out-of-state tuition. For the non-linear covariates, the etimated curve function for room and board suggest that increasing room and board costs were associated with greater out-of-state tution but with plateus at \$3,000, \$4,000, and \$5,000 before the relationship turns positive again. The functional curve of instructional expenditure per student resembled an S-curve that levels out at approximately \$20,000 per student.

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

```
coefi = coef(college.fwd, id=8)
test.mat = model.matrix(Outstate ~ ., data = College[-train,])

#predictions
college.fwd.preds = test.mat[,names(coefi)]%*%coefi
college.gam.preds = predict(college.gam, newdata = college.test)
rmse(college.fwd.preds,college.test$Outstate)
```

```
## [1] 2124.964
```

```
rmse(college.gam.preds, college.test$Outstate)
```

```
## [1] 1877.393
```

We find that the test set root mean squared error for the generalized additive model (2013) is better than the foward subset selection model (2134).

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

As mention in part (b), the results for Room.Board and Expend suggest a non-linear relationship.

11. In Section 7.7, it was mentioned that GAMs are generally fit using a backfitting approach. The idea behind backfitting is actually quite simple. We will now explore backfitting in the context of multiple linear regression.

(a) Generate a response Y and two predictors X_1 and X_2, with n = 100.

```
set.seed(30)
X.1 = rnorm(100, 0,3)
X.2 = rnorm(100, 0,3)
e = rnorm(100,0,1)
Y = -3*X.1 + 1.5*X.2 + e
```

(b) Initialize $\hat{\beta}_1$ to take on a value of your choice. It does not matter what value you choose.

```
beta1.hat = 1
```

(c) Keeping $\hat{\beta}_1$ fixed, fit the model $Y - \hat{\beta}_1 X_1 = \beta_0 + \beta_2 X_2 + \epsilon$.

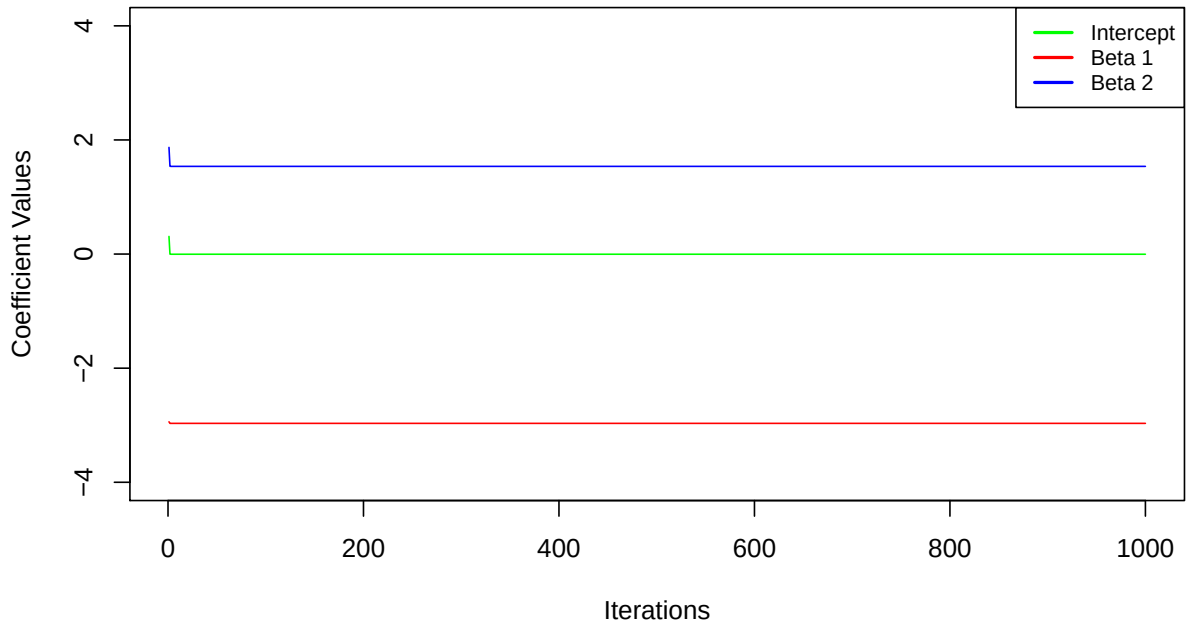
```
beta2.hat = lm( (Y - beta1.hat*X.1) ~ X.2 )$coef[2]
```

(d) Keeping $\hat{\beta}_2$ fixed, fit the model $Y - \hat{\beta}_2 X_2 = \beta_0 + \beta_1 X_1 + \epsilon$.

```
beta1.hat = lm( (Y - beta2.hat*X.2) ~ X.2 )$coef[2]
```

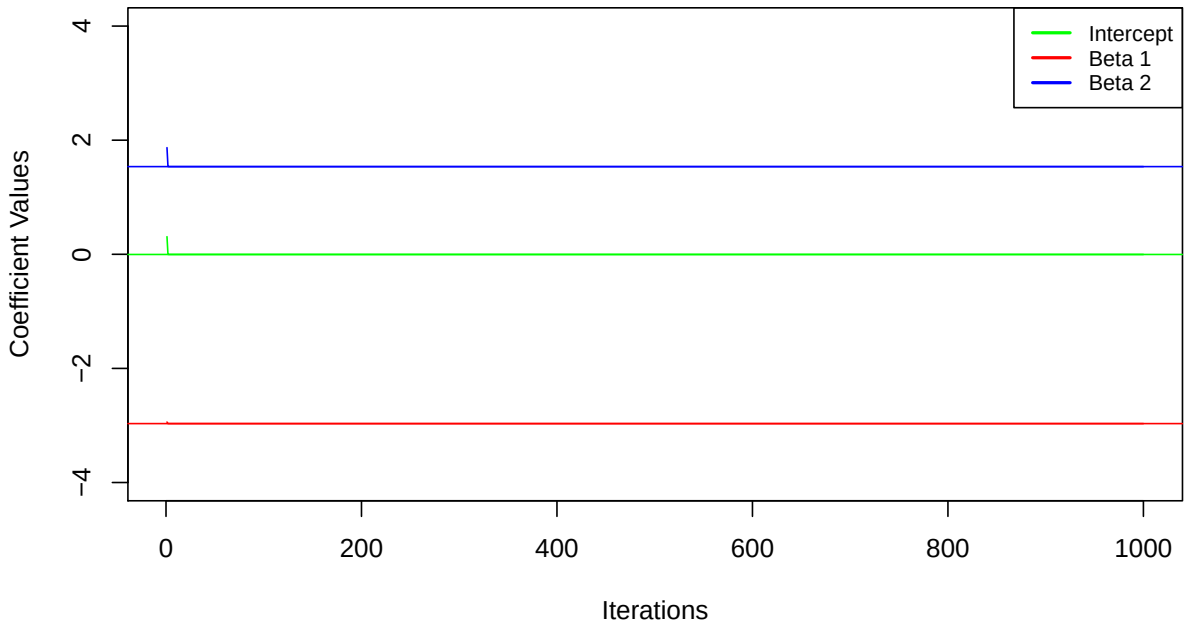
(e) Write a for loop to repeat (c) and (d) 1,000 times. Report the estimates of $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ at each iteration of the for loop. Create a plot in which each of these values is displayed, with $\hat{\beta}_0$, $\hat{\beta}_1$, and $\hat{\beta}_2$ each shown in a different color.

```r
sims = 1000
beta.hat = matrix(NA, nrow = 3, ncol = 1000)
beta1.hat = 1
for (i in 1:sims){
  beta.hat[3,i] = beta2.hat = lm( (Y - beta1.hat*X.1) ~ X.2 )$coef[2]
  beta.hat[2,i] = beta1.hat = lm( (Y - beta2.hat*X.2) ~ X.1 )$coef[2]
  beta.hat[1,i] = lm( (Y - beta1.hat*X.1 - beta2.hat*X.2) ~ 1 )$coef[1]
}
plot(1:sims,beta.hat[1,], type="l", col = "green",
     ylim = c(-4,4),
     xlab = "Iterations",
     ylab = "Coefficient Values")
lines(1:sims,beta.hat[2,], type="l", col = "red")
lines(1:sims,beta.hat[3,], type="l", col = "blue")
legend("topright", legend = c("Intercept",
                              "Beta 1",
                              "Beta 2"),
       col = c("green",
               "red",
               "blue"),
       lty=1,
       lwd=2,
       cex = 0.8)
```



(f) Compare your answer in (e) to the results of simply performing multiple linear regression to predict Y using X_1 and X_2. Use the abline() function to overlay those multiple linear regression coefficient estimates on the plot obtained in (e).

```r
#multiple linear regression
fit = lm(Y ~ X.1 + X.2)

plot(1:sims,beta.hat[1,], type="l", col = "green",
     ylim = c(-4,4),
     xlab = "Iterations",
     ylab = "Coefficient Values")
lines(1:sims,beta.hat[2,], type="l", col = "red")
lines(1:sims,beta.hat[3,], type="l", col = "blue")
abline(h = fit$coefficients[1], col = "green")
abline(h = fit$coefficients[2], col = "red")
abline(h = fit$coefficients[3], col = "blue")
legend("topright", legend = c("Intercept",
                              "Beta 1",
                              "Beta 2"),
       col = c("green",
               "red",
               "blue"),
       lty=1,
       lwd=2,
       cex = 0.8)
```



(g) On this data set, how many backfitting iterations were required in order to obtain a "good" approximation to the multiple regression coefficient estimates?

```r
which.max(beta.hat[1,] < 0.1 & beta.hat[1,] > -0.1)
```

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

```
which.max(beta.hat[2,] < -2.9 & beta.hat[2,] > -3.1)
```

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

```
which.max(beta.hat[3,] < 1.6 & beta.hat[2,] > 1.4)
```

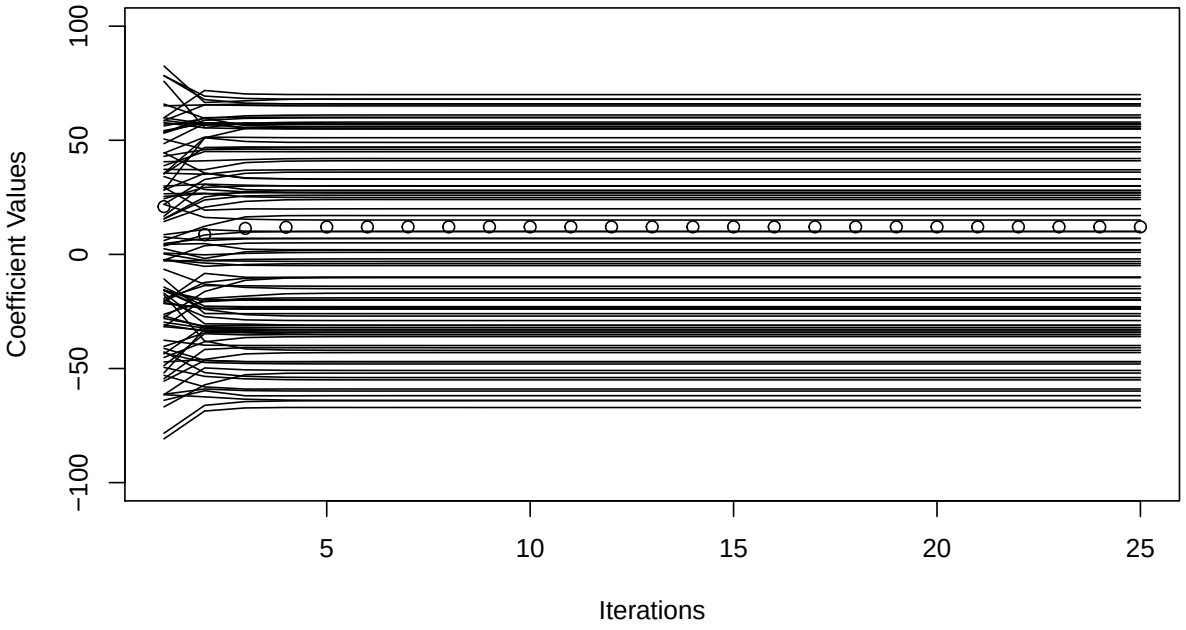
```
## [1] 1
```

The backfitting estimations were within 0.1 of the true value after only 2 iterations

12. This problem is a continuation of the previous exercise. In a toy example with p = 100, show that one can approximate the multiple linear regression coefficient estimates by repeatedly performing simple linear regression in a backfitting procedure. How many backfitting iterations are required in order to obtain a "good" approximation to the multiple regression coefficient estimates? Create a plot to justify your answer.

```
library(mvtnorm)
#creating p=100 and n=1,000 with Y response
set.seed(4509)
X = rmvnorm(1000, mean = rep(0,100))
beta.true = sample(-70:70, size = 100, replace = TRUE)
e = rnorm(1000, 0, 2)
Y = X%*%beta.true + e

#backfitting
sims = 25
beta.hat = matrix(NA, nrow = 100, ncol = sims)
beta.hat[2:100,1] = 1
for (i in 1:sims){
  for (j in 1:100){
    beta.hat[j,i] = lm( Y - X[,-j]%*%beta.hat[-j,i] ~ X[,j])$coef[2]
  }
  if (i < sims){beta.hat[2:100,i+1] = beta.hat[2:100,i]}
  else {break}
}
```

```
plot(1:sims, beta.hat[1,],
     ylim = c(-100, 100),
     xlab = "Iterations",
     ylab = "Coefficient Values")
for (i in 2:100){
  lines(1:sims, beta.hat[i,])
}
```



Convergence to approximate estimates of the true beta values appears to occur within 5 iterations.