ISL - Chapter 7 Exercises
Moving Beyond Linearity

An introduction to Statistical Learning, with Applications in R
- G. James, D. Witten, T. Hastie, R. Tibshirani

Thu Nguyen

10 July, 2019

Contents

```
library(ISLR)
library(boot)
```

Exercise 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.
(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.

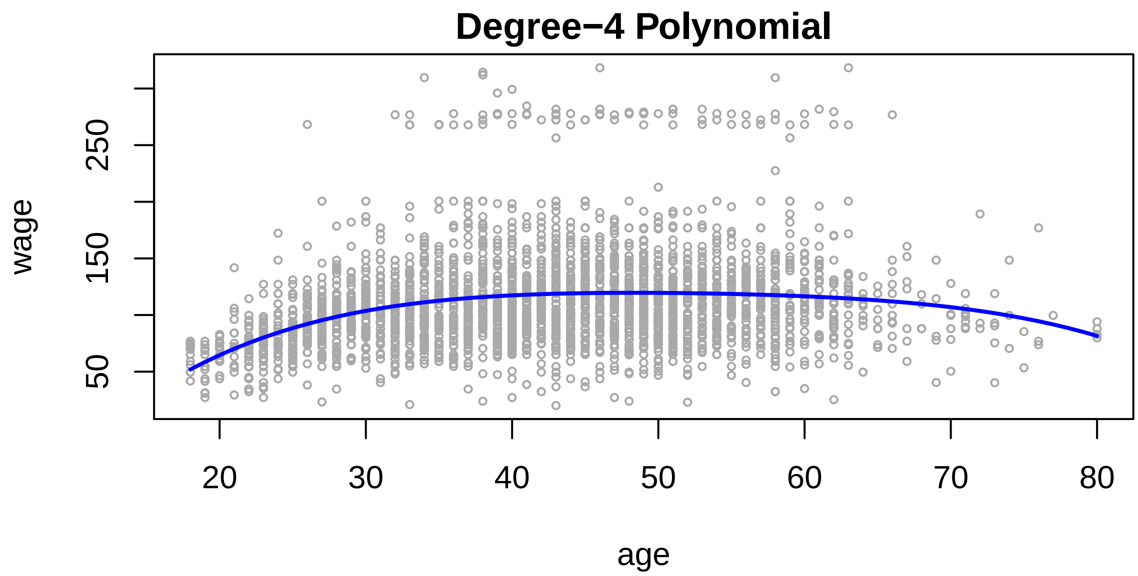
(a) Polynomial regression with 10-fold cross-validation

```
attach(Wage)
set.seed(1)
cv.error.10 <- rep(0, 10)
for (i in 1:10) {
  fit <- glm(wage ~ poly(age, i), data = Wage)
  cv.error.10[i] <- round(cv.glm(Wage, fit, K = 10)$delta[1], 2)
}
cv.deg <- which.min(cv.error.10)
t(data.frame(Degree = 1:10, MSE = cv.error.10))
```

```
##           [,1]    [,2]   [,3]    [,4]    [,5]    [,6]   [,7]    [,8]    [,9]   [,10]
## Degree    1.00    2.00    3.0    4.00    5.00    6.00    7.0    8.00    9.00   10.00
## MSE    1675.84 1601.01 1598.8 1594.22 1594.63 1594.89 1595.5 1595.44 1596.34 1595.83
```

As seen from the summary table of MSE on the entire set, a polynomial of degree 4 returns a best fit.

```r
par(mar=c(4,4,1.5,.5))
agelims <- range(age)
age.grid <- seq(agelims[1], agelims[2])
fit <- glm(wage ~ poly(age, 4), data = Wage)
pred <- predict(fit, newdata = list(age = age.grid))
plot(age, wage, xlim = agelims, cex = .5, col = 'darkgrey')
title('Degree-4 Polynomial')
lines(age.grid, pred, lwd = 2, col = 'blue')
```



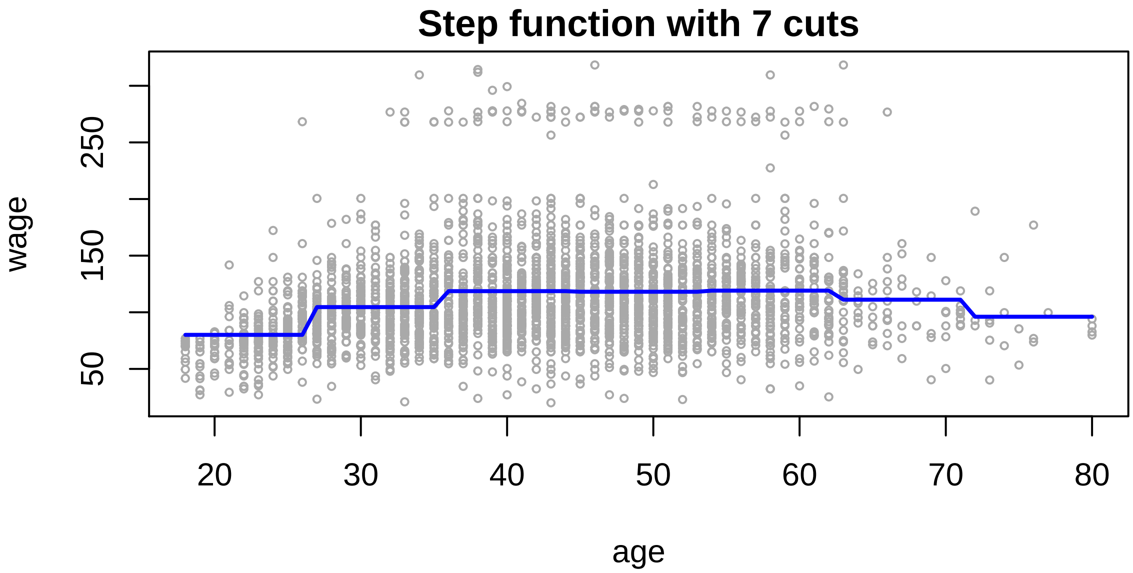
(b) Step function with 10-fold cross-validation

```r
set.seed(1)
cv.error.10 <- rep(0, 10)
for (i in 2:10) {
  Wage$temp <- cut(age, i)
  fit <- glm(wage ~ temp, data = Wage)
  cv.error.10[i] <- cv.glm(Wage, fit, K = 10)$delta[1]
}
cv.deg <- which.min(cv.error.10[-1])
t(data.frame(Step_cut = 2:11, MSE = cv.error.10))
```

```
##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]    [,10]
## Step_cut    2    3.000    4.000    5.000    6.000    7.000    8.000    9.000   10.000   11.000
## MSE         0 1733.968 1683.398 1639.253 1631.339 1623.162 1612.098 1600.689 1611.707 1605.738
```

As seen from the summary table of MSE on the entire set, a step function with 7 cuts returns a best fit.

```r
par(mar=c(4,4,1.5,.5))
fit <- glm(wage ~ cut(age, cv.deg), data = Wage)
pred <- predict(fit, newdata = list(age = age.grid))
plot(age, wage, xlim = agelims, cex = .5, col = 'darkgrey')
title('Step function with 7 cuts')
lines(age.grid, pred, lwd = 2, col = 'blue')
```


Step function with 7 cuts
wage
50
150
250
20
30
40
50
60
70
80
age

■

Exercise 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.
(b) lot the polynomial fits for a range of different polynomial degrees (say, from 1 to 10), and report the associated residual sum of squares.
(c) Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.
(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.
(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.
(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

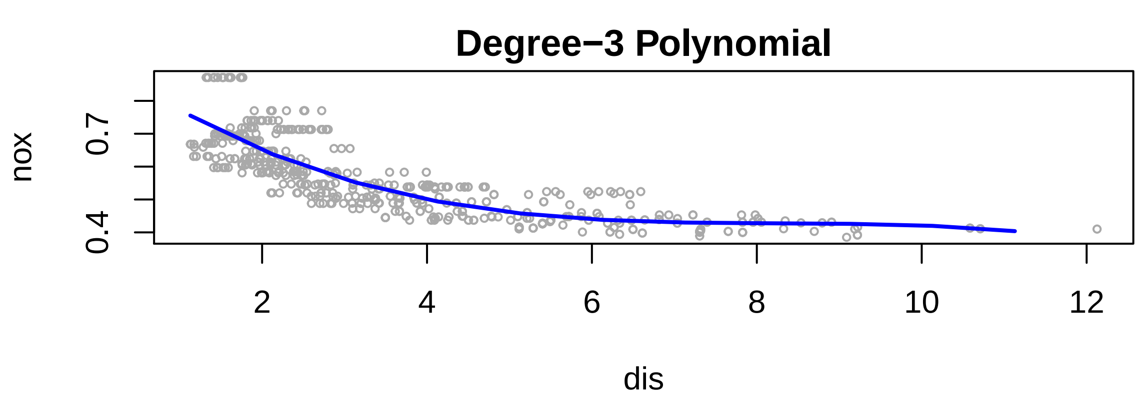
---

(a) Polynomial regression of `nox ~ poly(dis, 3)`

```r
library(MASS)
attach(Boston)
dislims <- range(dis)
dis.grid <- seq(dislims[1], dislims[2])
(fit <- glm(nox ~ poly(dis, 3), data = Boston))
```

```
##
## Call:  glm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Coefficients:
##   (Intercept)  poly(dis, 3)1  poly(dis, 3)2  poly(dis, 3)3
##        0.5547        -2.0031         0.8563        -0.3180
##
## Degrees of Freedom: 505 Total (i.e. Null);  502 Residual
## Null Deviance:       6.781
## Residual Deviance: 1.934     AIC: -1371
```

```r
par(mar=c(4,4,1.5,.5))
pred <- predict(fit, newdata = list(dis = dis.grid))
plot(dis, nox, xlim = dislims, cex = .5, col = 'darkgrey')
title('Degree-3 Polynomial')
lines(dis.grid, pred, lwd = 2, col = 'blue')
```



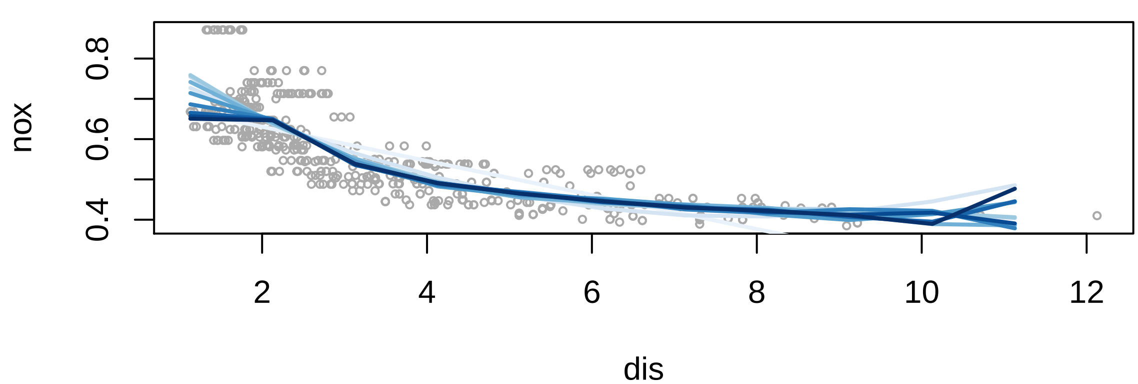
(b) Polynomial regression of degree in $1:10$

```r
mse <- rep(0, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  mse[i] <- round(mean((pred - nox)^2),4)
}
mse.deg <- which.min(cv.error.10)
t(data.frame(Degree = 1:10, MSE = mse))
```

```
##          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]   [,10]
## Degree 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000 9.0000 10.0000
## MSE    0.0413 0.0263 0.0283 0.0285 0.0294 0.0269 0.0272 0.0259 0.0265  0.0252
```

As seen from the summary table of MSE on the entire set, a polynomial of degree 1 returns a best fit.

```r
par(mar=c(4,4,.5,.5))
library(RColorBrewer)
mycols <- colorRampPalette(brewer.pal(9,'Blues'))(30)
plot(dis, nox, xlim = dislims, cex = .5, col = 'darkgrey')
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  lines(dis.grid, pred, lwd = 2, col = mycols[3*i])
}
```



(c) Polynomial regression with 10-fold cross-validation

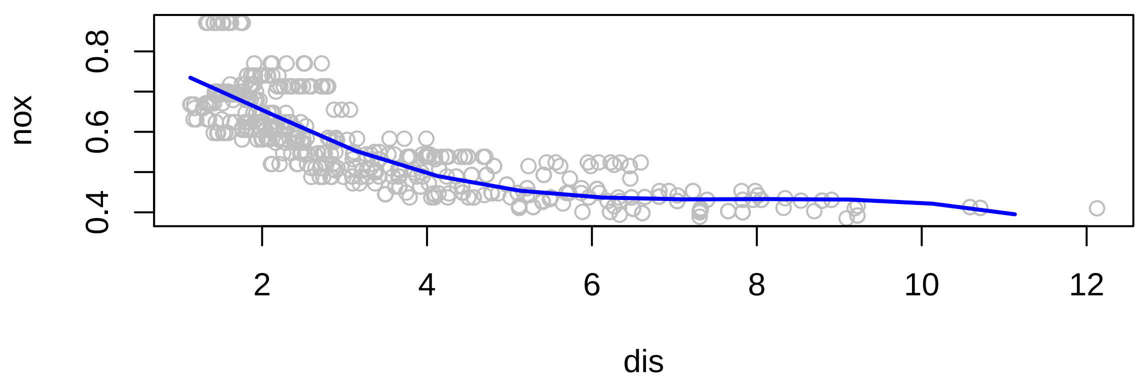
```r
set.seed(1)
cv.error.10 <- rep(0, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  cv.error.10[i] <- round(cv.glm(Boston, fit, K = 10)$delta[1], 4)
}
cv.deg <- which.min(cv.error.10)
t(data.frame(Degree = 1:10, MSE = cv.error.10))
```

```
##          [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]   [,10]
## Degree 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 7.0000 8.0000 9.0000 10.0000
## MSE    0.0055 0.0041 0.0039 0.0039 0.0043 0.0051 0.0137 0.0053 0.0134  0.0041
```

As seen, a polynomial of degree 3 returns a best fit, which is similar to what was returned via the MSE approach.

(d) Regression spline at $df = 4$

```
par(mar=c(4,4,.5,.5))
library(splines)
fit <- lm(nox ~ bs(dis, df=4), data = Boston)
pred <- predict(fit, newdata = list(dis = dis.grid))
plot(dis, nox, col = 'gray')
lines(dis.grid, pred, lwd = 2, col = 'blue')
```



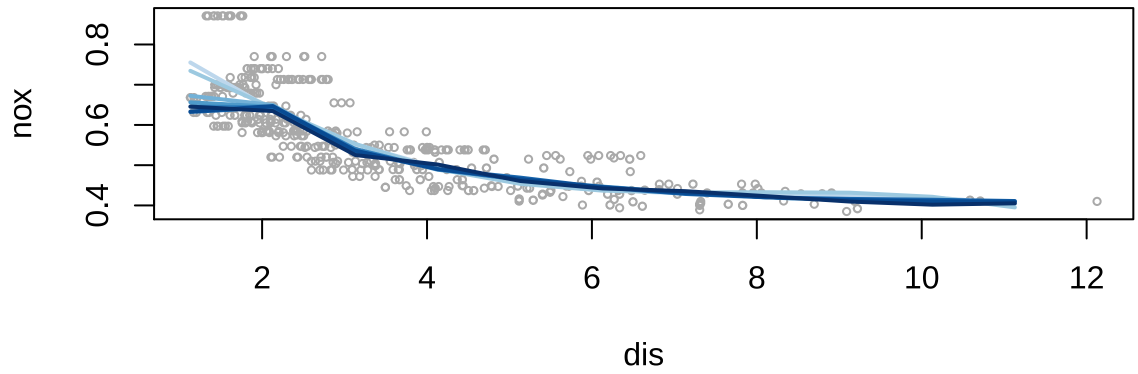
(e) Regression spline over different df

```
rss <- rep(0, 10)
for (i in 1:10) {
  fit <- lm(nox ~ bs(dis, df=i), data = Boston)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  rss[i] <- round((pred - nox)^2,4)
}
rss.deg <- which.min(rss)
t(data.frame(DF = 1:10, RSS = rss))
```

```
##       [,1]   [,2]   [,3]   [,4]   [,5]  [,6]   [,7]   [,8]   [,9]   [,10]
## DF  1.0000 2.0000 3.0000 4.0000 5.0000 6.000 7.0000 8.0000 9.0000 10.0000
## RSS 0.0472 0.0472 0.0472 0.0386 0.0181 0.014 0.0116 0.0089 0.0091  0.0116
```

As seen, a regression spline with $df = 8$ returns a best fit.

```
par(mar=c(4,4,.5,.5))
plot(dis, nox, xlim = dislims, cex = .5, col = 'darkgrey')
for (i in 1:10) {
  fit <- lm(nox ~ bs(dis, df=i), data = Boston)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  lines(dis.grid, pred, lwd = 2, col = mycols[3*i])
}
```



(f) Regression spline with 10-fold cross-validation

```
attach(Wage)
set.seed(1)
cv.error.10 <- rep(0, 10)
for (i in 1:10) {
  fit <- glm(nox ~ bs(dis, df=i), data = Boston)
  cv.error.10[i] <- round(cv.glm(Boston, fit, K = 10)$delta[1], 6)
}
cv.deg <- which.min(cv.error.10)
t(data.frame(DF = 1:10, MSE = cv.error.10))
```

```
##         [,1]     [,2]   [,3]     [,4]     [,5]     [,6]     [,7]     [,8]     [,9]     [,10]
## DF  1.000000 2.000000 3.0000 4.000000 5.000000 6.000000 7.000000 8.000000 9.000000 10.000000
## MSE 0.003866 0.003887 0.0039 0.003862 0.003699 0.003715 0.003694 0.003715 0.003733  0.003655
```

As seen, a regression spline with $df = 10$ returns a best fit.

■

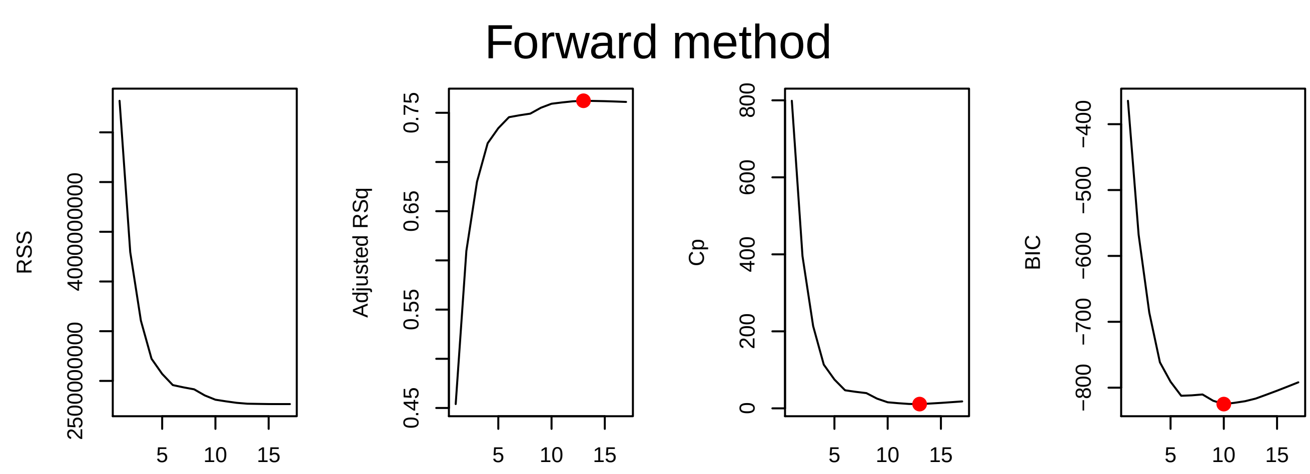
Exercise 10

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.
(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.
(c) Evaluate the model obtained on the test set, and explain the results obtained.
(d) For which variables, if any, is there evidence of a non-linear relationship with the response?

(a) Forward stepwise subset selection

```r
library(leaps)
attach(College)
set.seed(1)
idx <- sample(nrow(College), round(nrow(College)*.8,0), replace = FALSE)
train <- College[idx,]
test <- College[-idx,]
regfit.fwd <- regsubsets(Outstate ~ ., data = train, method = 'forward', nvmax = ncol(College))
reg.fwd.sum <- summary(regfit.fwd)
par(mfrow=c(1,4), oma = c(0, 0, 2, 0)); par(mar=c(3,5,1,1))
plot(reg.fwd.sum$rss, xlab = 'Number of Variables', ylab = 'RSS', type = 'l')
plot(reg.fwd.sum$adjr2, xlab = 'Number of Variables', ylab = 'Adjusted RSq', type = 'l')
points(which.max(reg.fwd.sum$adjr2), reg.fwd.sum$adjr2[which.max(reg.fwd.sum$adjr2)],
col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$cp, xlab = 'Number of Variables', ylab = 'Cp', type = 'l')
points(which.min(reg.fwd.sum$cp), reg.fwd.sum$cp[which.min(reg.fwd.sum$cp)],
col = 'red', cex = 2, pch = 20)
plot(reg.fwd.sum$bic, xlab = 'Number of Variables', ylab = 'BIC', type = 'l')
points(which.min(reg.fwd.sum$bic), reg.fwd.sum$bic[which.min(reg.fwd.sum$bic)],
col = 'red', cex = 2, pch = 20)
mtext('Forward method', outer = TRUE, cex = 1.5)
```



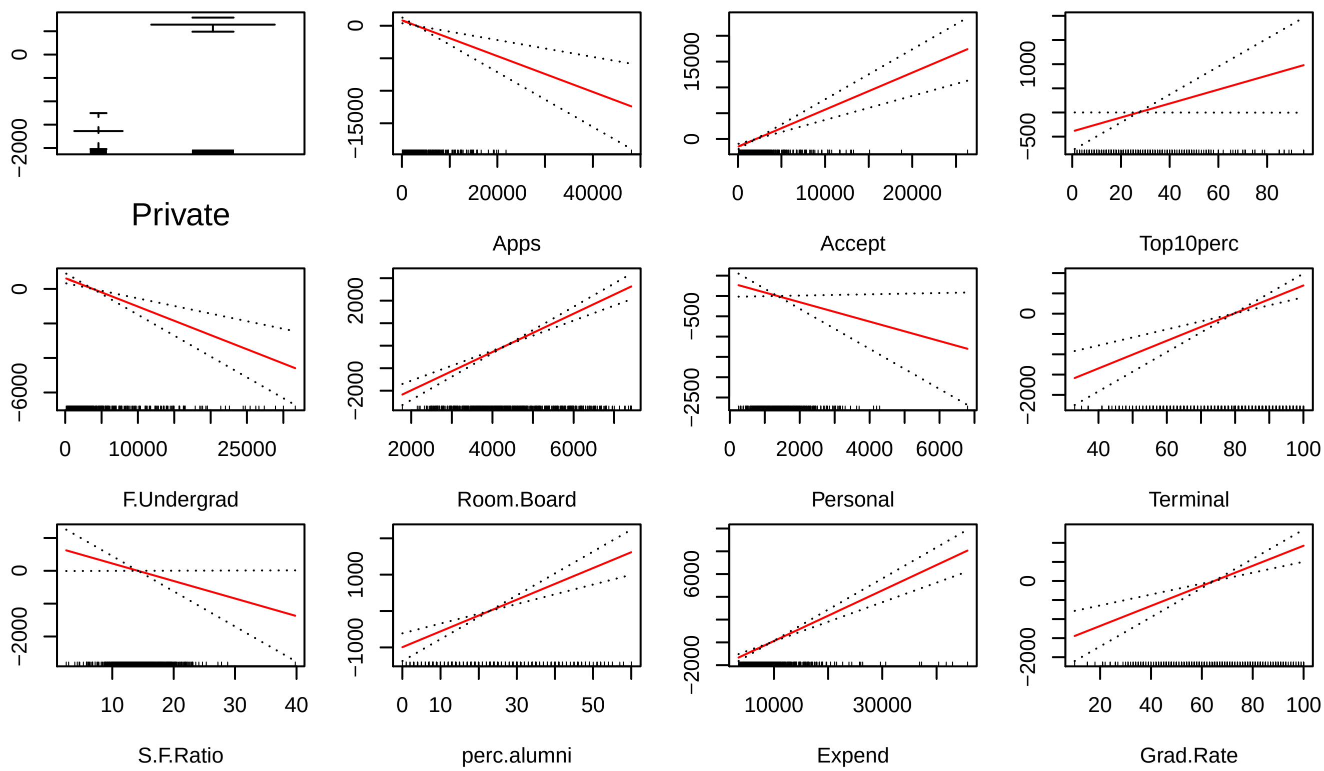
As seen from plots, a reasonable choice subset selection would have 12 variables + a constant:

```r
as.matrix(coef(regfit.fwd, id = 12))
```

```
##                      [,1]
## (Intercept) -1840.2756379
## PrivateYes   2278.4149597
## Apps           -0.2754579
## Accept          0.7182791
## Top10perc      14.4382168
## F.Undergrad    -0.1649786
## Room.Board      0.8508526
## Personal       -0.2392541
## Terminal       34.0339036
## S.F.Ratio     -53.3834424
## perc.alumni    43.5388417
## Expend          0.2227041
## Grad.Rate      26.3290944
```

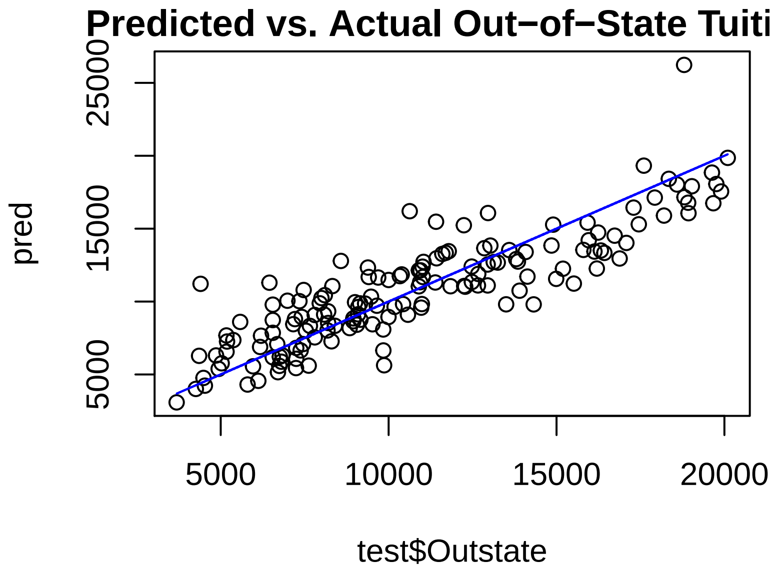
(b) GAM fit with selected variables from (a)

```r
par(mfrow=c(3,4), mar=c(4,3,.5,.5))
library(gam)
fit <- gam(Outstate ~ . - Enroll - Top25perc - P.Undergrad - Books - PhD, data = train)
plot.Gam(fit, se=TRUE, col = 'red')
```



(c) Model evaluation

```
par(mar=c(4,4,1.5,.5))
pred <- predict(fit, newdata = test)
plot(test$Outstate, pred, main = 'Predicted vs. Actual Out-of-State Tuition')
lines(x = test$Outstate, y = test$Outstate, col = 'blue')
```



(f) Linear vs. Non-linear relationship with the response

```
summary(fit)
```

```
##
## Call: gam(formula = Outstate ~ . - Enroll - Top25perc - P.Undergrad -
##     Books - PhD, data = train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -6990.58 -1307.01   -91.06  1274.51  9517.48
##
## (Dispersion Parameter for gaussian family taken to be 3743275)
##
##     Null Deviance: 9754133995 on 621 degrees of freedom
## Residual Deviance: 2279654318 on 609 degrees of freedom
## AIC: 11194.28
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                Df     Sum Sq    Mean Sq  F value                Pr(>F)
## Private         1 3112251568 3112251568 831.4248 < 0.00000000000000022 ***
## Apps            1  964450697  964450697 257.6489 < 0.00000000000000022 ***
## Accept          1   29027328   29027328   7.7545             0.0055244 **
## Top10perc       1 1113746601 1113746601 297.5327 < 0.00000000000000022 ***
## F.Undergrad     1  275493664  275493664  73.5970 < 0.00000000000000022 ***
## Room.Board      1  997841318  997841318 266.5691 < 0.00000000000000022 ***
## Personal        1   41850055   41850055  11.1801             0.0008775 ***
## Terminal        1  246925915  246925915  65.9652  0.000000000000002558 ***
```

```
## S.F.Ratio     1  172800503  172800503  46.1629  0.000000000025951472 ***
## perc.alumni   1  183855869  183855869  49.1163  0.000000000006417870 ***
## Expend        1  264198257  264198257  70.5794  0.000000000000000312 ***
## Grad.Rate     1   72037903   72037903  19.2446  0.000013551093973274 ***
## Residuals   609 2279654318    3743275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

As seen from the summary table, since all the p-values are all much smaller than .05, all of the 12 selected variables would be better suited with non-linear functions with respect to the response.

■