Lab 7: Moving Beyond Linearity

ACTL3142 and ACTL5110

Questions

Conceptual Questions

1. ⋆ (ISLR2, Q7.2) Suppose that a curve $\hat{g}$ is computed to smoothly fit a set of n points using the following formula:

$$\hat{g} = \arg\min_{g} \left(\sum_{i}^{n} (y_i - g(x_i))^2 + \lambda \int \left[g^{(m)}(x)\right]^2 \mathrm{d}x \right),$$

   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. $\lambda = \infty$, $m = 0$.

   b. $\lambda = \infty$, $m = 1$.

   c. $\lambda = \infty$, $m = 2$.

   d. $\lambda = \infty$, $m = 3$.

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

   Solution

2. ⋆ (ISLR2, Q7.3) 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.

   Solution

3. (ISLR2, Q7.4) Suppose we fit a curve with basis functions

$$b_1(X) = I(0 \leq X \leq 2) - (X-1)I(1 \leq X \leq 2),$$

$$b_2(X) = (X-3)I(3 \leq X \leq 4) + I(4 < X \leq 5).$$

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 $X = 6$. Note the intercepts, slopes, and other relevant information.

Solution

4. ⋆ (ISLR2, Q7.5) Consider two curves, $\hat{g}_1$ and $\hat{g}_2$, defined by

$$\hat{g}_1 = \arg\min_g \left(\sum_i^n (y_i - g(x_i))^2 + \lambda \int \left[g^{(3)}(x) \right]^2 \mathrm{d}x \right),$$

$$\hat{g}_2 = \arg\min_g \left(\sum_i^n (y_i - g(x_i))^2 + \lambda \int \left[g^{(4)}(x) \right]^2 \mathrm{d}x \right),$$

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?

b. As $\lambda \to \infty$ will $\hat{g}_1$ or $\hat{g}_2$ have the smaller test RSS?

c. For $\lambda = 0$, will $\hat{g}_1$ or $\hat{g}_2$ have the smaller training and test RSS?

Solution

Applied Questions

1. (ISLR2, Q7.6) 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 crossvalidation to choose the optimal number of cuts. Make a plot of the fit obtained.

Solution

2. (ISLR2, Q7.9) ⋆ 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. Plot 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.

   Solution

Solutions

Conceptual Questions

1. a. Since $\lambda \to \infty$, *any* non-zero value of $\int (g^{(m)}(x))^2 \mathrm{d}x$ will lead to an infinite penalty. Hence, it must integrate to zero, which can only occur if $g^{(m)}(x) = 0$. For $m = 0$, this will occur at $g(x) = 0$.

   b. The optimisation will find the best curve that fits the data provided $g^{(m)}(x) = 0$. So, if $m = 1$, $g(x) = k$, where k is the mean of the y_is.

   c. $g(x)$ will be the straight line of best fit for the data

   d. $g(x)$ will the be quadratic (parabola) of best fit for the data

   e. Since $\lambda = 0$, the smoothness penalty is ignored. Hence g will be the polynomial that goes through all the data points.

2. The sketch should look like:

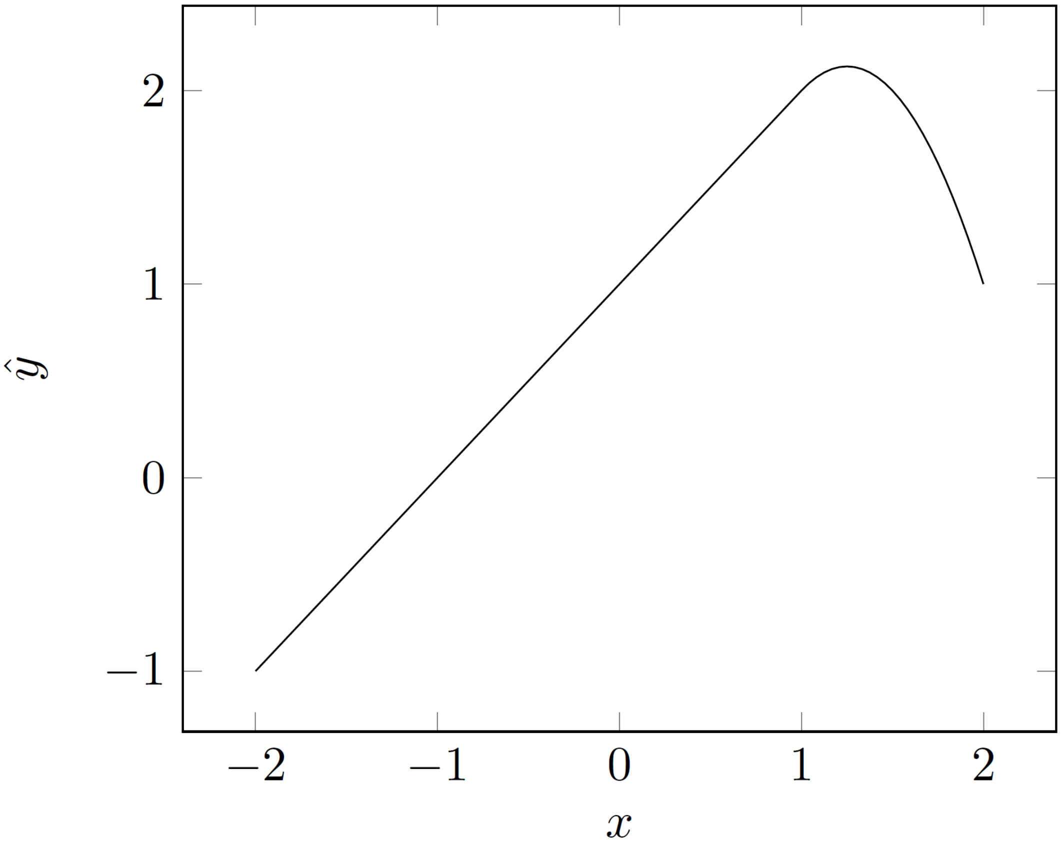


Figure 1: Solution

3. The sketch should look like:

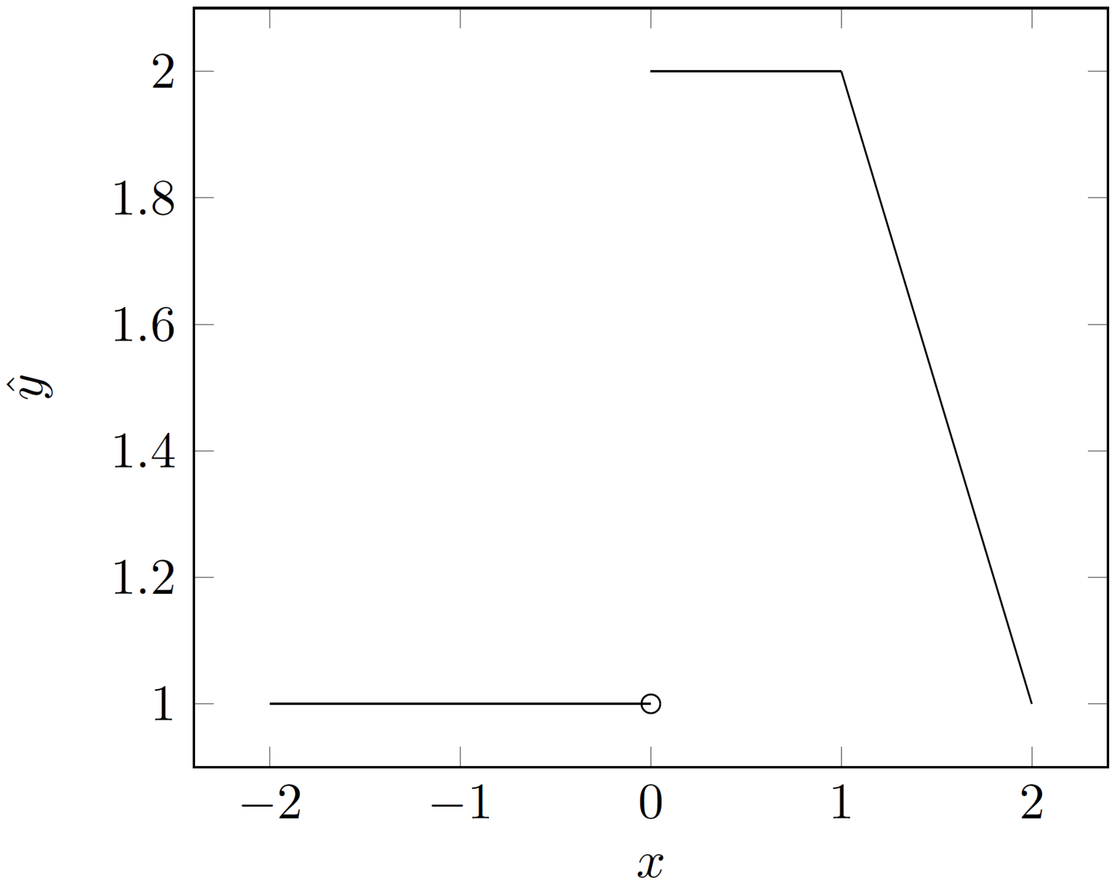


Figure 2: Solution

4. a. $\hat{g}_2$ will correspond to the cubic polynomial of best fit for the data and $\hat{g}_1$ to the quadratic (parabola) of best fit for the data. Therefore, since a cubic polynomial is more flexible than a cuadratic polynomial, $\hat{g}_2$ should have the better training RSS

   b. $\hat{g}_1$, conventionally, since it is not as likely to overfit the data. However, $\hat{g}_2$ may outperform it if the underlying trend is more cubic than quadratic.

   c. If $\lambda = 0$, then both optimisations are doing the same thing, so neither one is better or worse than the other.

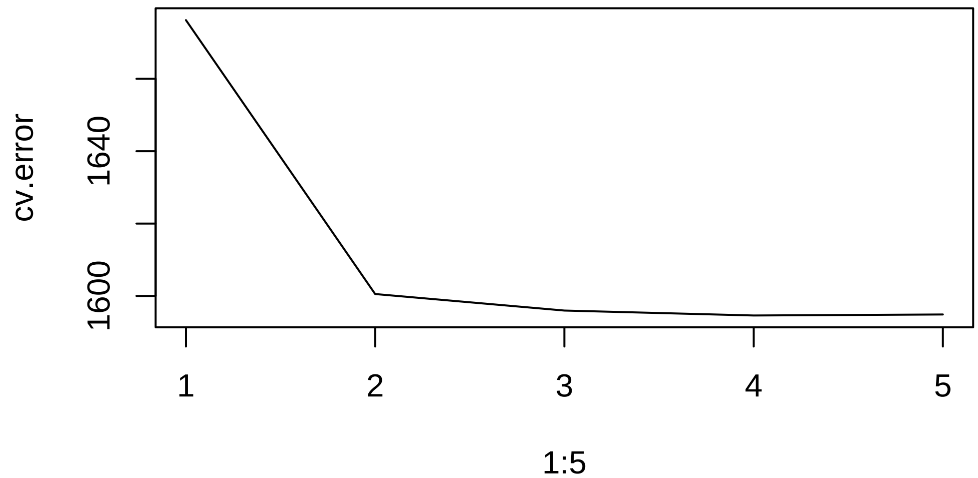
Applied Questions

1. a.
```
library(ISLR2)
library(boot)
```

```r
fit1 <- glm(wage ~ age, data = Wage)
# try using an iterative (and somewhat naive) ANOVA method
degree <- 1
while (tail(summary(fit1)$coefficients[, 4], 1) < 0.05) {
  degree <- degree + 1
  fit1 <- glm(wage ~ poly(age, degree, raw = TRUE), data = Wage)
}
```

According to this method, the best fit is when we use degree 3. Let us try CV as the question suggests.

```r
set.seed(1)
cv.error <- rep(0, 5)
for (i in 1:5) {
  fit1 <- glm(wage ~ poly(age, i, raw = TRUE), data = Wage)
  cv.error[i] <- cv.glm(Wage, fit1)$delta[1]
}
plot(1:5, cv.error, type = "l")
```

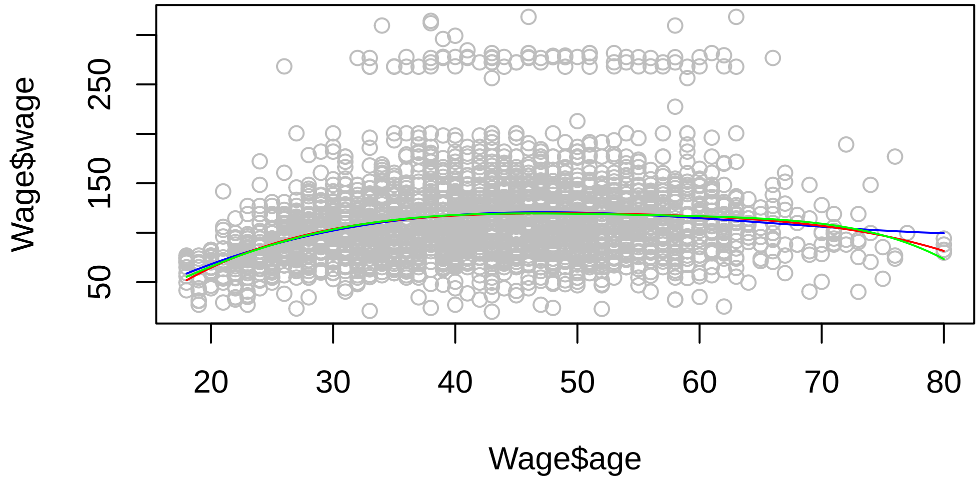


It seems like the best choice is between degrees 3, 4 and 5.

```r
fit3 <- lm(wage ~ poly(age, 3, raw = TRUE), data = Wage)
fit4 <- lm(wage ~ poly(age, 4, raw = TRUE), data = Wage)
fit5 <- lm(wage ~ poly(age, 5, raw = TRUE), data = Wage)
age.grid <- data.frame(age = seq(from = 18, to = 80, length.out = 100))
pred3 <- predict(fit3, age.grid)
pred4 <- predict(fit4, age.grid)
pred5 <- predict(fit5, age.grid)
```

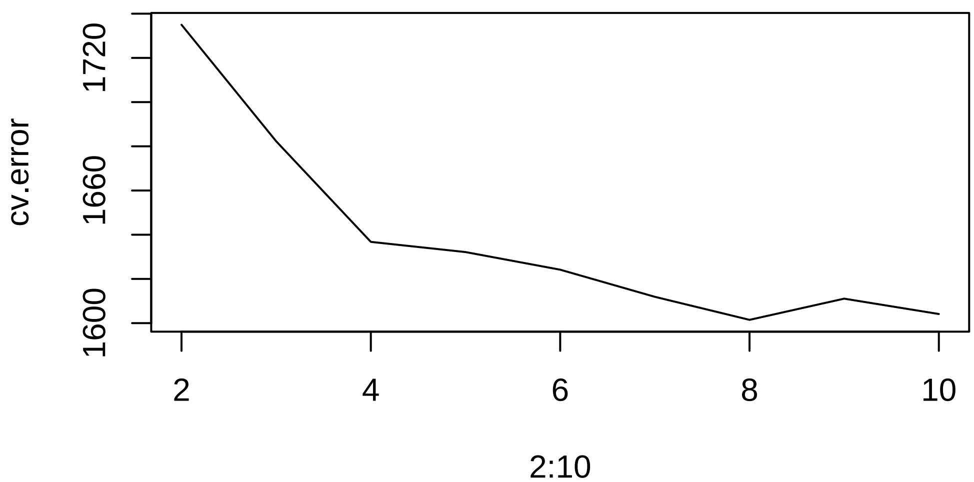
```r
library(ISLR2)
plot(Wage$age, Wage$wage, col = "grey")
```

```r
lines(age.grid$age, pred3, col = "blue")
lines(age.grid$age, pred4, col = "red")
lines(age.grid$age, pred5, col = "green")
```



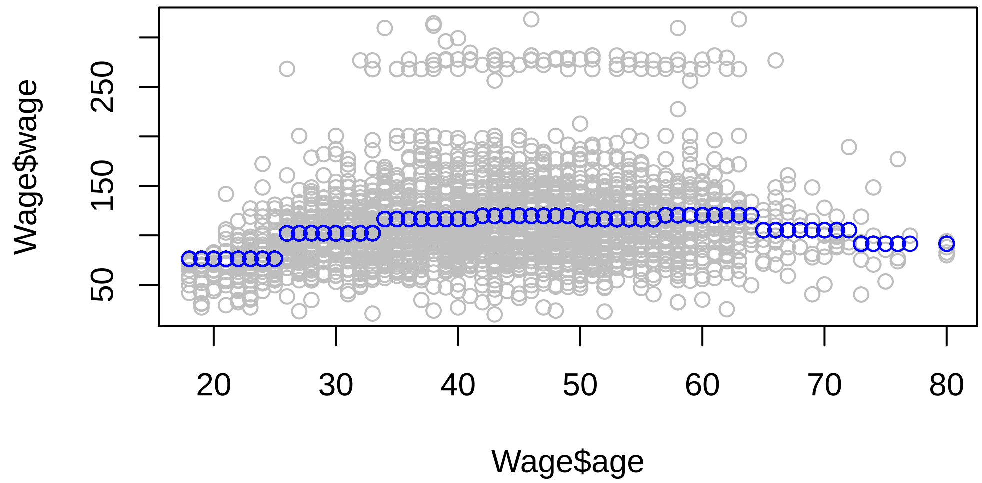
The plots are clearly very close to each other.

b.
```r
cv.error <- rep(0, 9)
myWage <- Wage
for (i in 2:10) {
  myWage["age.fact"] <- cut(myWage$age, i)
  fit <- glm(wage ~ age.fact, data = myWage)
  cv.error[i - 1] <- cv.glm(myWage, fit, K = 10)$delta[1]
}
plot(2:10, cv.error, type = "l")
```

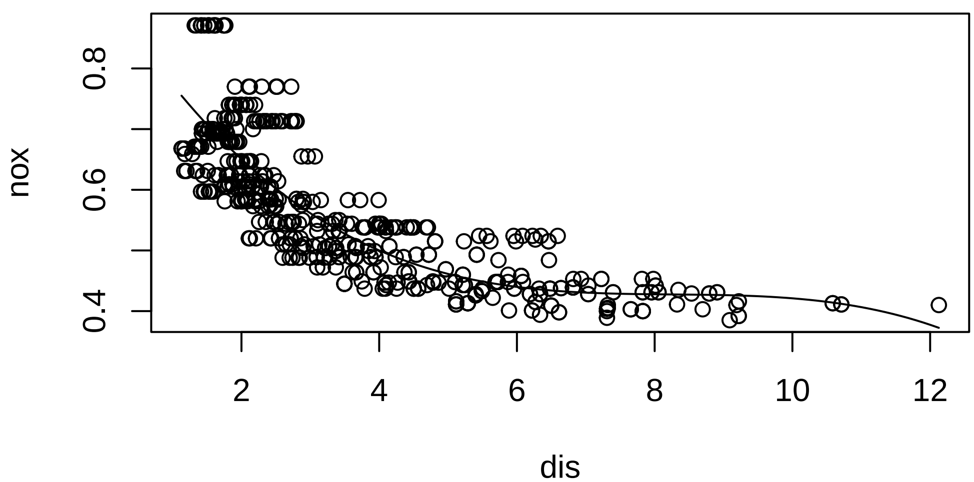


It seems like 8 cut-points is appropriate here.

```r
myWage["age.fact"] <- cut(myWage$age, 8)
fit <- glm(wage ~ age.fact, data = myWage)
plot(Wage$age, Wage$wage, col = "grey")
points(myWage$age, fit$fitted.values, col = "blue")
```

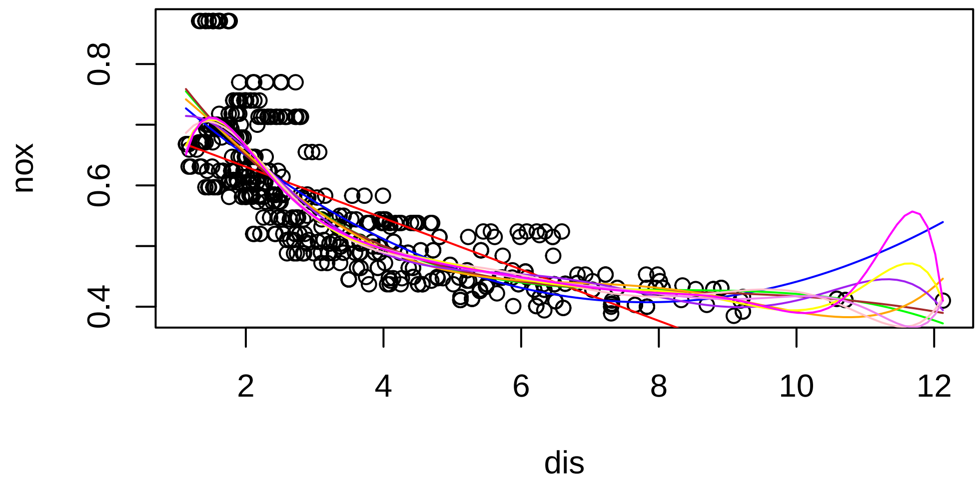


2. a.
```r
library(ISLR2)
fit <- lm(nox ~ poly(dis, 3, raw = TRUE), data = Boston)
dis.grid <- seq(min(Boston$dis), max(Boston$dis), length.out = 100)
pred <- predict(fit, newdata = list(dis = dis.grid))
plot(Boston$dis, Boston$nox, xlab = "dis", ylab = "nox")
lines(dis.grid, pred)
```



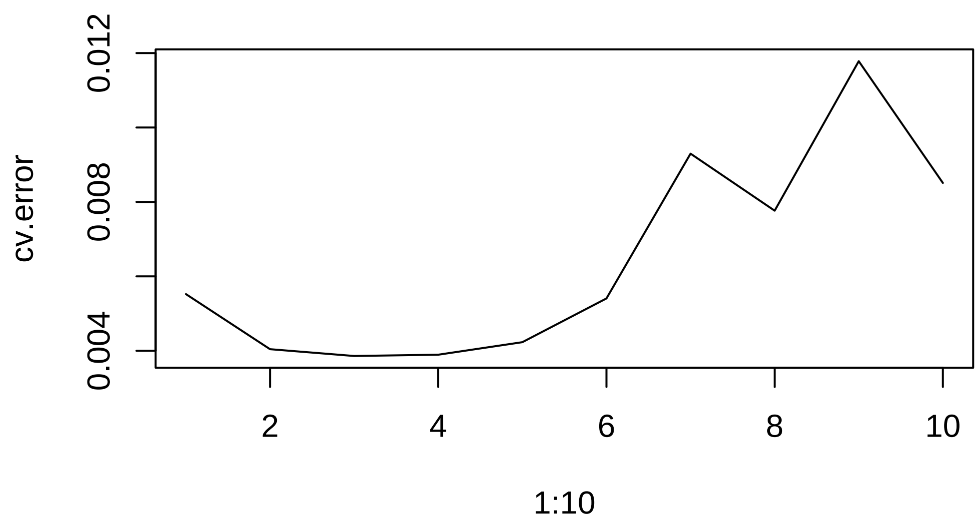
b.
```r
rss <- rep(0, 10)
colours <- c(
  "red", "blue", "green", "brown", "orange", "purple",
  "pink", "yellow", "violet", "magenta"
```

```
)
plot(Boston$dis, Boston$nox, xlab = "dis", ylab = "nox")
for (i in 1:10) {
  fit <- lm(nox ~ poly(dis, i, raw = TRUE), data = Boston)
  rss[i] <- sum(fit$residuals^2)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  lines(dis.grid, pred, col = colours[i])
}
```



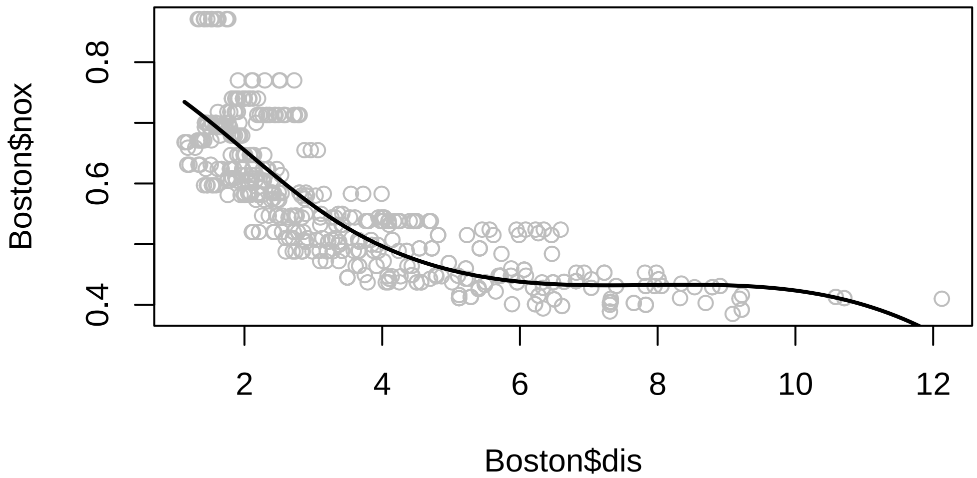
c.

```
cv.error <- rep(0, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i, raw = TRUE), data = Boston)
  cv.error[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(1:10, cv.error, type = "l")
```

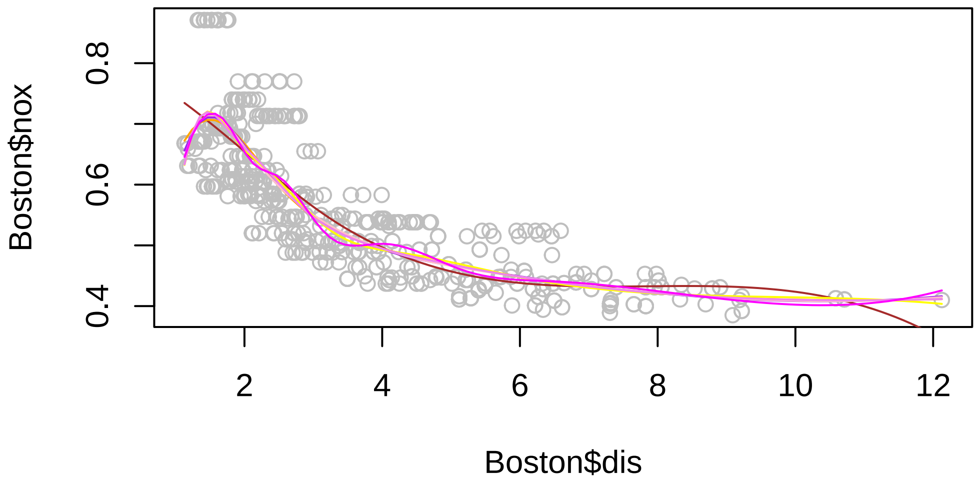


It looks like a degree of 3 is best.

d.
```
library(splines)
fit <- lm(nox ~ bs(dis, df = 4), data = Boston)
# note that by default, the bs function does not include
# the intercept term in the basis generated
pred <- predict(fit, newdata = list(dis = dis.grid))
plot(Boston$dis, Boston$nox, col = "gray")
lines(dis.grid, pred, lwd = 2)
```



e.
```
plot(Boston$dis, Boston$nox, col = "gray")
rss <- rep(0, 10)
for (i in 4:13) {
  fit <- lm(nox ~ bs(dis, df = i), data = Boston)
  rss[i - 3] <- sum(fit$residuals^2)
  pred <- predict(fit, newdata = list(dis = dis.grid))
  lines(dis.grid, pred, col = colours[i])
}
```



f.
```
set.seed(1)
cv.error <- rep(0, 10)
options(warn=-1)
for (i in 4:13) {
  fit <- glm(nox ~ bs(dis, df = i), data = Boston)
  cv.error[i - 3] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
options(warn=0)  # reset to default
plot(4:13, cv.error, type = "l", xlab = "df")
```