Nonlinear Modeling I, Solutions

Splines and Polynomial Regression (close to chapter 7, ISLR)

M Loecher

1. Exercises
   (a) Compare fits using `poly` with "manual" polynomials
   (b) Use `anova` to find the best degree
   (c) Produce an analogous plot for the *Auto* data.

```
attach(Wage)
fit <- lm(wage ~ poly(age, 4), data = Wage)
coef(summary(fit))
```

```
            Estimate Std. Error    t value     Pr(>|t|)
```

(Intercept) 111.70361 0.7287409 153.283015 0.000000e+00 poly(age, 4)1 447.06785 39.9147851 11.200558 1.484604e-28 poly(age, 4)2 -478.31581 39.9147851 -11.983424 2.355831e-32 poly(age, 4)3 125.52169 39.9147851 3.144742 1.678622e-03 poly(age, 4)4 -77.91118 39.9147851 -1.951938 5.103865e-02

```
fit2 <- lm(wage ~ poly(age, 4, raw =TRUE), data = Wage)
coef(summary(fit2))
```

```
                               Estimate   Std. Error   t value
```

(Intercept) -1.841542e+02 6.004038e+01 -3.067172 poly(age, 4, raw = TRUE)1 2.124552e+01 5.886748e+00 3.609042 poly(age, 4, raw = TRUE)2 -5.638593e-01 2.061083e-01 -2.735743 poly(age, 4, raw = TRUE)3 6.810688e-03 3.065931e-03 2.221409 poly(age, 4, raw = TRUE)4 -3.203830e-05 1.641359e-05 -1.951938 Pr(>|t|) (Intercept) 0.0021802539 poly(age, 4, raw = TRUE)1 0.0003123618 poly(age, 4, raw = TRUE)2 0.0062606446 poly(age, 4, raw = TRUE)3 0.0263977518 poly(age, 4, raw = TRUE)4 0.0510386498

```
fit2a <- lm(wage ~ age + I(age^2) + I(age^3) + I(age^4), data = Wage)
coef(fit2a)
```

(Intercept) age I(age^2) I(age^3) I(age^4) -1.841542e+02 2.124552e+01 -5.638593e-01 6.810688e-03 -3.203830e-05

```
fit2b <- lm(wage ~ cbind(age, age^2, age^3, age^4), data = Wage)
coef(fit2b)
```

```
                  (Intercept) cbind(age, age^2, age^3, age^4)age
                -1.841542e+02                       2.124552e+01
```

cbind(age, age^2, age^3, age^4) cbind(age, age^2, age^3, age^4) -5.638593e-01 6.810688e-03 cbind(age, age^2, age^3, age^4) -3.203830e-05

```
agelims <- range(age)
age.grid <- seq(from = agelims[1], to = agelims[2])
preds <- predict(fit, newdata = list(age = age.grid), se = TRUE)


preds2 <- predict(fit2, newdata = list(age = age.grid), se = TRUE)
max(abs(preds$fit - preds2$fit))
```

[1] 7.81597e-11

```
fit.1 <- lm(wage ~ age, data = Wage)
fit.2 <- lm(wage ~ poly(age, 2), data = Wage)
fit.3 <- lm(wage ~ poly(age, 3), data = Wage)
fit.4 <- lm(wage ~ poly(age, 4), data = Wage)
fit.5 <- lm(wage ~ poly(age, 5), data = Wage)
anova(fit.1, fit.2, fit.3, fit.4, fit.5)
```

Analysis of Variance Table

Model 1: wage ~ age Model 2: wage ~ poly(age, 2) Model 3: wage ~ poly(age, 3) Model 4: wage ~ poly(age, 4) Model 5: wage ~ poly(age, 5)
Res.Df RSS Df Sum of Sq F Pr(>F)
1 2998 5022216
2 2997 4793430 1 228786 143.5931 < 2.2e-16 * **3 2996 4777674 1 15756 9.8888 0.001679** 4 2995 4771604 1 6070 3.8098 0.051046 .
5 2994 4770322 1 1283 0.8050 0.369682
— Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ‘’ 1

```
coef(summary(fit.5))
```

```
              Estimate Std. Error     t value      Pr(>|t|)
```

(Intercept) 111.70361 0.7287647 153.2780243 0.000000e+00 poly(age, 5)1 447.06785 39.9160847 11.2001930 1.491111e-28 poly(age, 5)2 -478.31581 39.9160847 -11.9830341 2.367734e-32 poly(age, 5)3 125.52169 39.9160847 3.1446392 1.679213e-03 poly(age, 5)4 -77.91118 39.9160847 -1.9518743 5.104623e-02 poly(age, 5)5 -35.81289 39.9160847 -0.8972045 3.696820e-01

```
fit.1 <- lm(wage ~ education + age, data = Wage)
fit.2 <- lm(wage ~ education + poly(age,2), data = Wage)
fit.3 <- lm(wage ~ education + poly(age,3), data = Wage)
anova(fit.1, fit.2, fit.3)
```

Analysis of Variance Table

Model 1: wage ~ education + age Model 2: wage ~ education + poly(age, 2) Model 3: wage ~ education + poly(age, 3) Res.Df RSS Df Sum of Sq F Pr(>F)
1 2994 3867992
2 2993 3725395 1 142597 114.6969 <2e-16 ** *3 2992 3719809 1 5587 4.4936 0.0341*
— Signif. codes: 0 '' ***0.001*** '' *0.01* '' 0.05 '.' 0.1 '' 1

2. Exercise

   Complete the 2nd plot

3. **Exercises, Coding**

   Fit cubic and natural cubic splines to the motorcycle data

   (a) with prespecified knots

   (b) with knots at uniform quantiles of the data

   (c) How would you decide on the optimal knots ?

4. **Exercises, Conceptual**

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 $\beta_0, \beta_1, \beta_2, \beta_3, \beta_4$.

For $x \leq \xi$, we have

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3,$$

so we take $a_1 = \beta_0$, $b_1 = \beta_1$, $c_1 = \beta_2$ and $d_1 = \beta_3$.

(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 piecewie polynomial.

For $x > \xi$, we have

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3 = (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\xi^2 \beta_4) x + (\beta_2 - 3\beta_4 \xi) x^2 + (\beta_3 + \beta_4) x^3,$$

so we take $a_2 = \beta_0 - \beta_4 \xi^3$, $b_2 = \beta_1 + 3\xi^2 \beta_4$, $c_2 = \beta_2 - 3\beta_4 \xi$ and $d_2 = \beta_3 + \beta_4$.

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

We have immediately that

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

and

$$f_2(\xi) = (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\xi^2 \beta_4) \xi + (\beta_2 - 3\beta_4 \xi) \xi^2 + (\beta_3 + \beta_4) \xi^3 = \beta_0 + \beta_1 \xi + \beta_2 \xi^2 + \beta_3 \xi^3.$$

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

We also have immediately that

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

and

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

(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.

We finally have that

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

and

$$f_2''(\xi) = 2(\beta_2 - 3\beta_4\xi) + 6(\beta_3 + \beta_4)\xi = 2\beta_2 + 6\beta_3\xi.$$

5. **Exercises, Conceptual**

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

$$\hat{g} = \arg\min_{g} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(m)}(x)]^2 dx \right),$$

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

In this case $\hat{g} = 0$ because a large smoothing parameter forces $g^{(0)}(x) \to 0$.

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

In this case $\hat{g} = c$ because a large smoothing parameter forces $g^{(1)}(x) \to 0$.

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

In this case $\hat{g} = cx + d$ because a large smoothing parameter forces $g^{(2)}(x) \to 0$.

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

In this case $\hat{g} = cx^2 + dx + e$ because a large smoothing parameter forces $g^{(3)}(x) \to 0$.

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

The penalty term doesn't play any role, so in this case g is the interpolating spline.

6. **Exercises, Conceptual**

consider two curves, $\hat{g}_1$ and $\hat{g}_2$, defined by

$$\hat{g}_1 = \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(3)}(x)]^2 dx \right)$$

$$\hat{g}_2 = \arg\min_g \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int [g^{(4)}(x)]^2 dx \right)$$

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 ?

The smoothing spline $\hat{g}_2$ will probably have the smaller training RSS because it will be a higher order polynomial due to the order of the penalty term (it will be more flexible).

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

As mentioned above we expect $\hat{g}_2$ to be more flexible, so it may overfit the data. It will probably be $\hat{g}_1$ that have the smaller test RSS.

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

If $\lambda = 0$, we have $\hat{g}_1 = \hat{g}_2$, so they will have the same training and test RSS.

7. **Exercises, Coding**

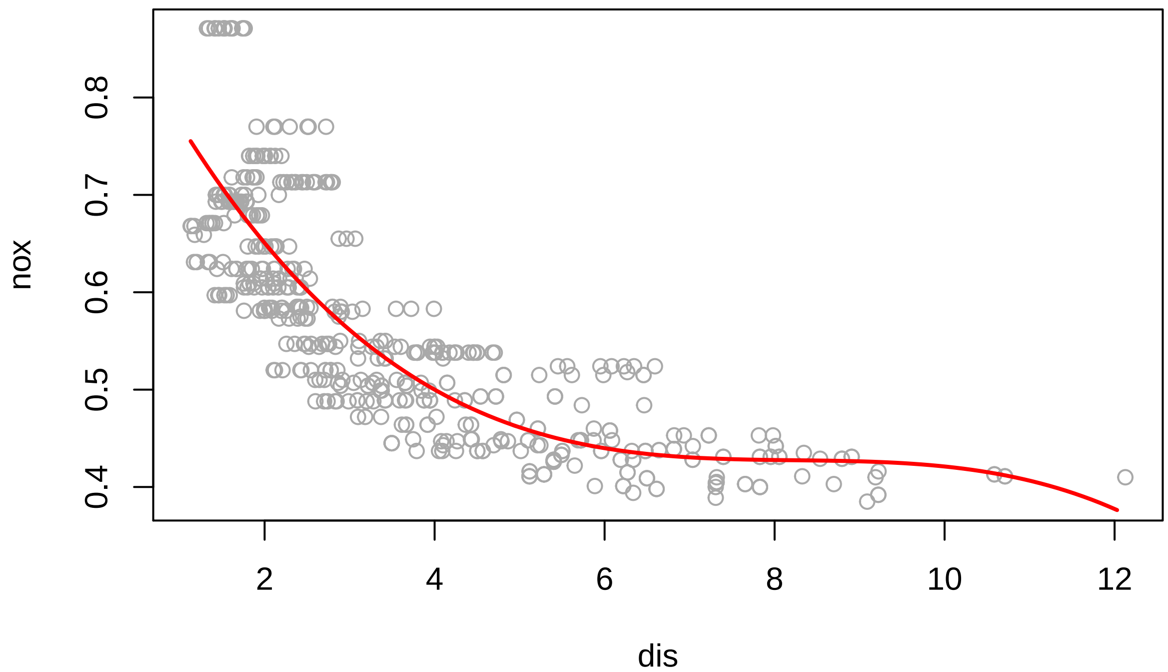
This question uses the variables "dis" (the weighted mean of distances to five Boston employment centers) and "nox" (nitrogen oxides concentration in parts per 10 million) from the "Boston" data. We will treat "dis" as the predictor and "nox" as the response.

(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
library(MASS)
set.seed(1)
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
```

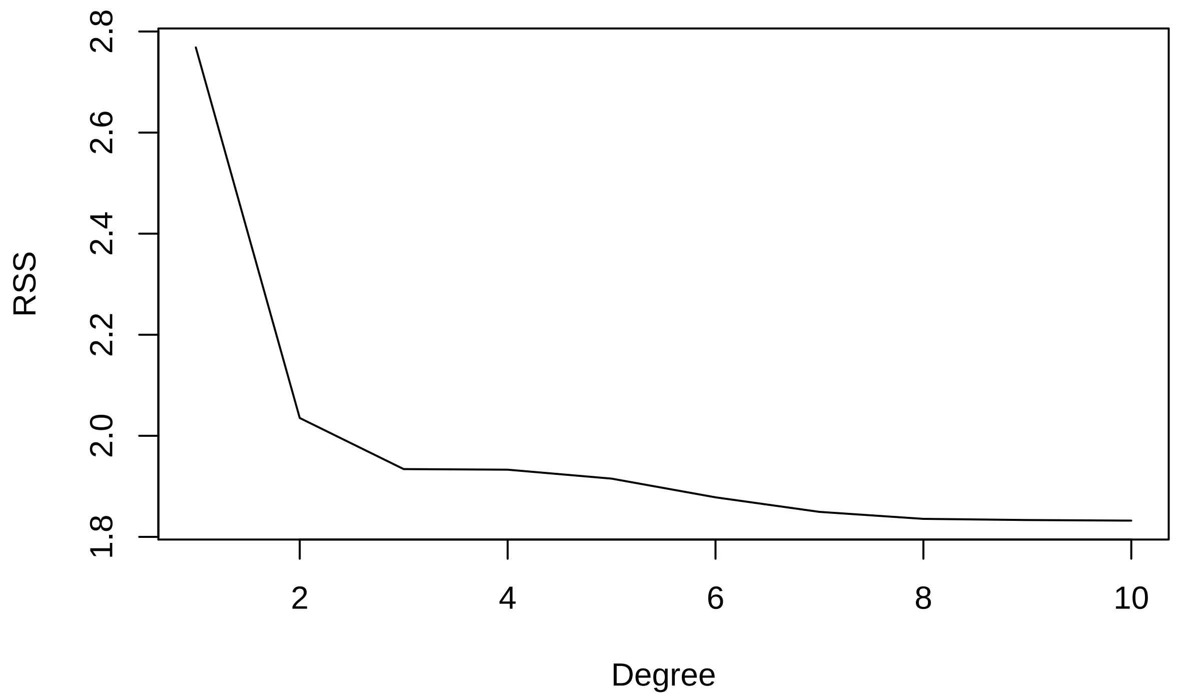
```r
dislims <- range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2], by = 0.1)
preds <- predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey")
lines(dis.grid, preds, col = "red", lwd = 2)
```



We may conclude 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.

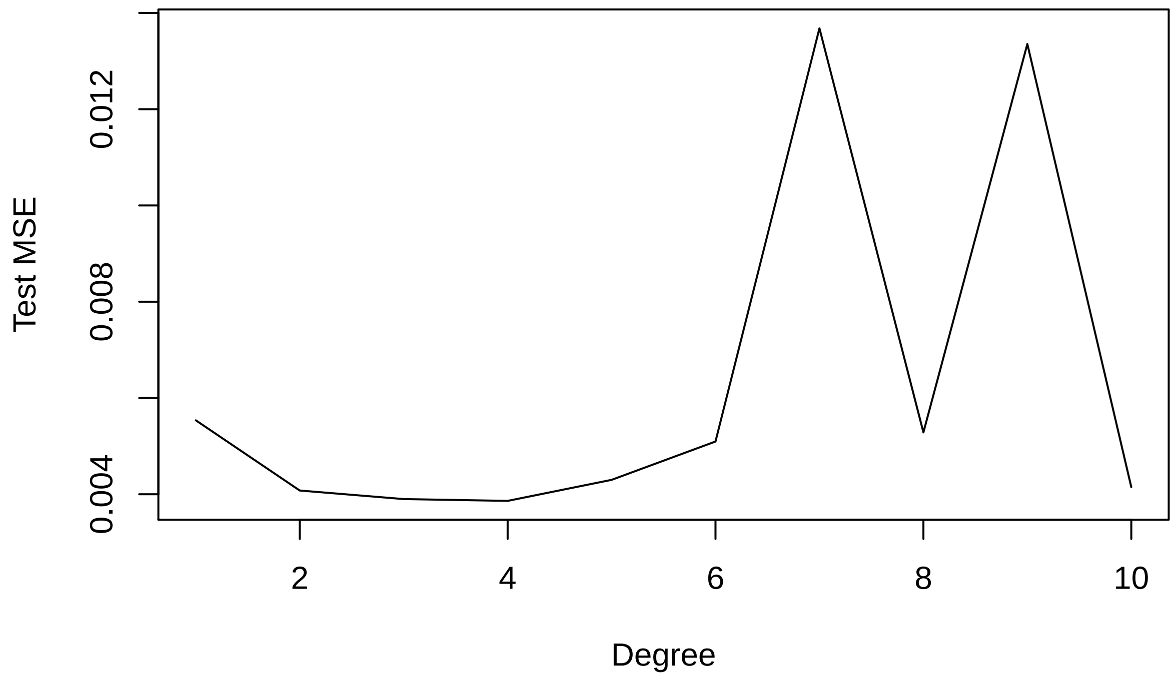
```r
rss <- rep(NA, 10)
for (i in 1:10) {
    fit <- lm(nox ~ poly(dis, i), data = Boston)
    rss[i] <- sum(fit$residuals^2)
}
plot(1:10, rss, xlab = "Degree", ylab = "RSS", type = "l")
```



It seems that the RSS decreases with the degree of the polynomial, and so is minimum for a polynomial of degree 10.

(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")
```



We may see that a polynomial of degree 4 minimizes the test MSE.

(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.

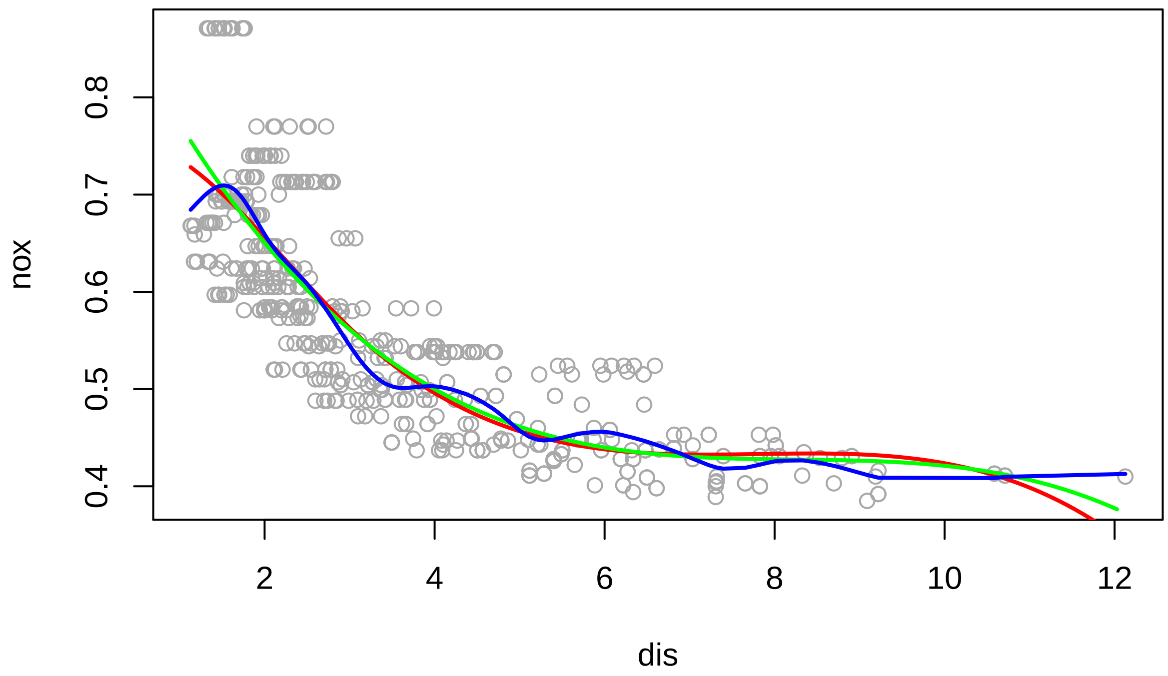
```
#fit2 <- lm(nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)
fit2 <- lm(nox ~ bs(dis, knots =3), data = Boston)
summary(fit2)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, knots = 3), data = Boston)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.126057 -0.039576 -0.008195  0.021148  0.193086
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.72823    0.01536  47.406  < 2e-16 ***
## bs(dis, knots = 3)1 -0.04034    0.02207  -1.828   0.0681 .
## bs(dis, knots = 3)2 -0.46757    0.02388 -19.578  < 2e-16 ***
## bs(dis, knots = 3)3 -0.18461    0.04355  -4.239 2.68e-05 ***
## bs(dis, knots = 3)4 -0.38593    0.04533  -8.513  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06187 on 501 degrees of freedom
## Multiple R-squared:  0.7172, Adjusted R-squared:  0.715
## F-statistic: 317.7 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
pred2 <- predict(fit2, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey")
lines(dis.grid, pred2, col = "red", lwd = 2)
lines(dis.grid, preds, col = "green", lwd = 2)

fit3 <- smooth.spline(Boston$dis, Boston$nox, cv = TRUE)
lines(fit3, col = "blue", lwd = 2)
```



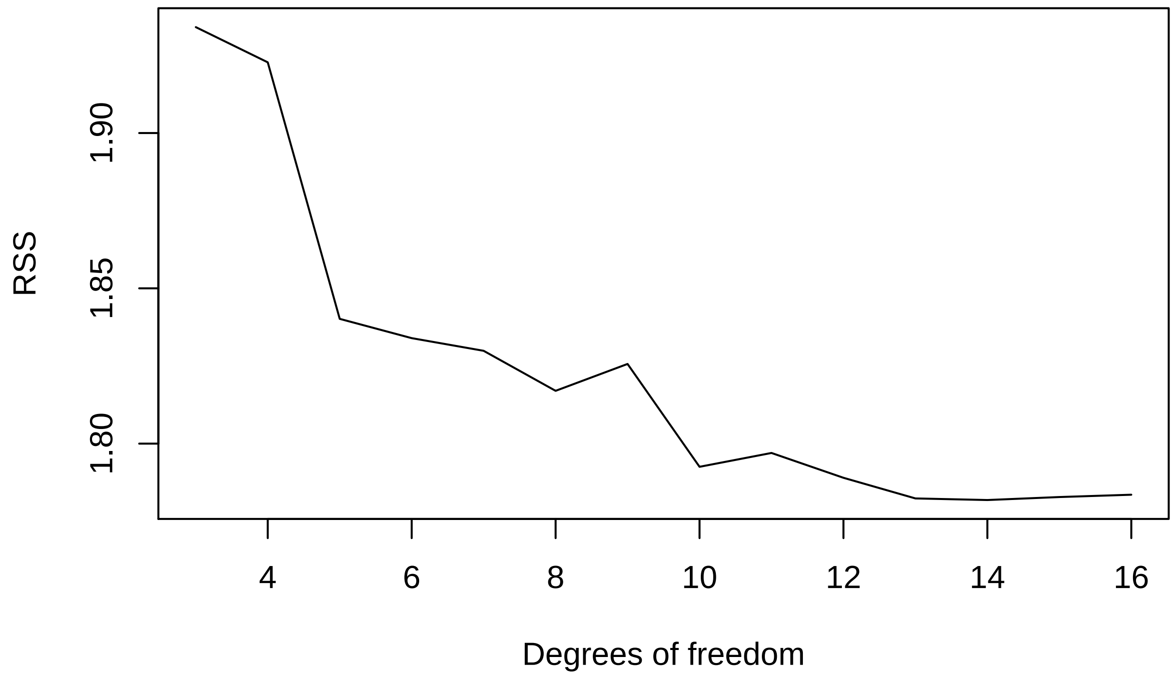

```
#fit3$df
#pred3 <- predict(fit3, list(dis = dis.grid))
#lines(dis.grid, pred3, col = "blue", lwd = 2)
```

We may conclude that all terms in spline fit are significant.

(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, 16)
for (i in 3:16) {
    fit <- lm(nox ~ bs(dis, df = i), data = Boston)
```

```r
    rss[i] <- sum(fit$residuals^2)
}
plot(3:16, rss[-c(1, 2)], xlab = "Degrees of freedom", ylab = "RSS", type = "l")
```

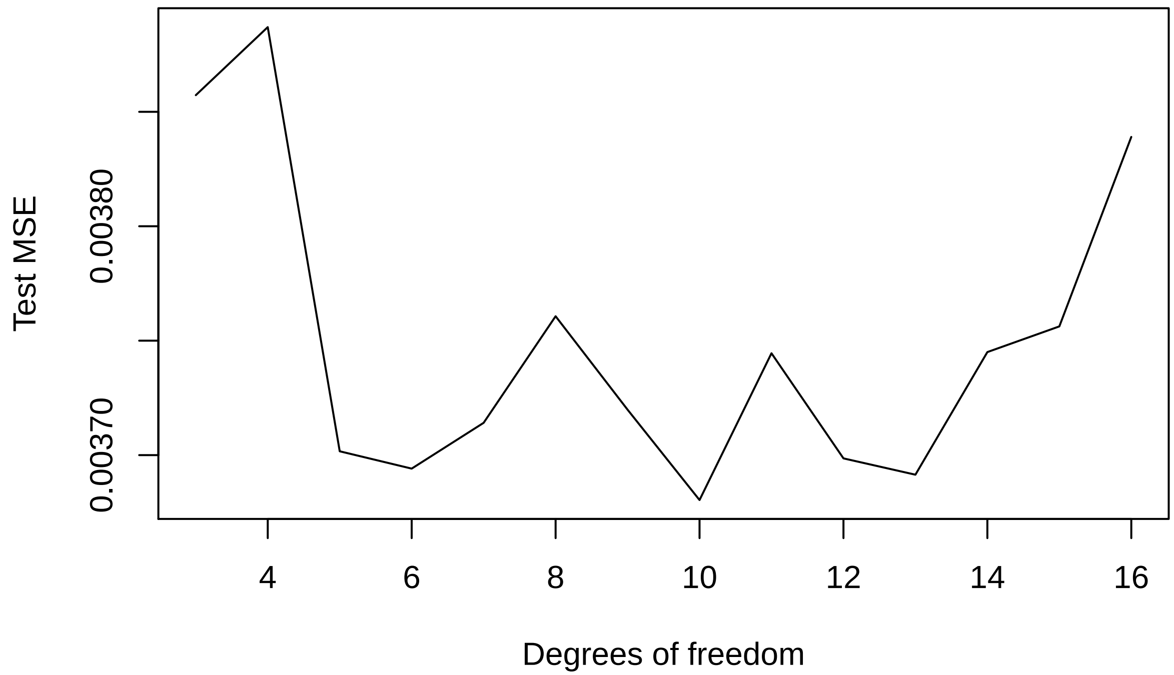


We may see that RSS decreases until 14 and then slightly increases after that.

(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.

```r
cv <- rep(NA, 16)
for (i in 3:16) {
    fit <- glm(nox ~ bs(dis, df = i), data = Boston)
```

```
    cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(3:16, cv[-c(1, 2)], xlab = "Degrees of freedom", ylab = "Test MSE", type = "l")
```



Test MSE is minimum for 10 degrees of freedom.