STAT 8330 FALL 2015 ASSIGNMENT 4

Peng Shao

September 27, 2015

▶ **Exercises 2. Solution.**

(a). For all $x \leq \xi$, $(x-\xi)_+^2 = 0$, then

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 \\ &= a_1 + b_1 x + c_1 x^2 + d_1 x^3 \end{aligned}$$

Thus,

$$\begin{aligned} a_1 &= \beta_0 \\ b_1 &= \beta_1 \\ c_1 &= \beta_2 \\ d_1 &= \beta_3 \end{aligned}$$

(b). For all $x > \xi$,

$$\begin{aligned} f(x) &= \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 (x-\xi)^3 \\ &= (\beta_0 - \beta_4 \xi^3) + (\beta_1 + 3\xi^2 \beta_4) x + (\beta_2 - 3\beta_4 \xi) x^2 + (\beta_3 + \beta_4) x^3 \\ &= a_1 + b_1 x + c_1 x^2 + d_1 x^3 \end{aligned}$$

Thus,

$$\begin{aligned} a_2 &= \beta_0 - \beta_4 \xi^3 \\ b_2 &= \beta_1 + 3\beta_4 \xi^2 \\ c_2 &= \beta_2 - 3\beta_4 \xi \\ d_2 &= \beta_3 + \beta_4 \end{aligned}$$

(c). Since

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

Then, $f_1(x) = f_2(x)$ at $x = \xi$, i.e. f(x) is continuous at ξ.

(d). Since

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

Then, $f_1'(x) = f_2'(x)$ at $x = \xi$, i.e. f'(x) is continuous at ξ.

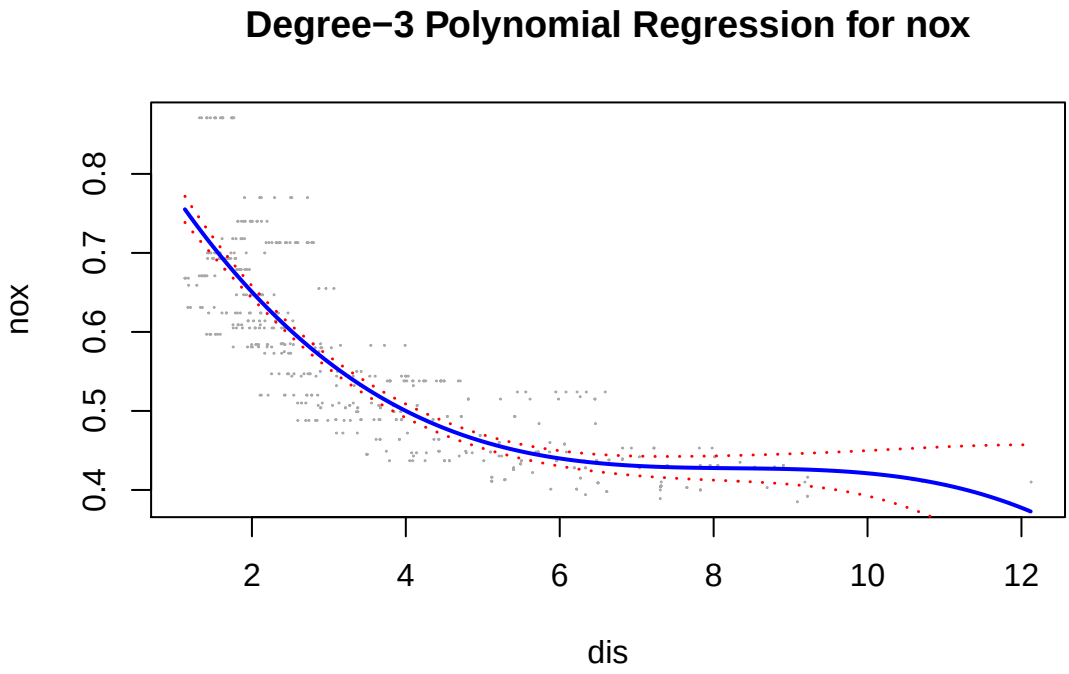
(e).

$$\begin{aligned} f_1''(\xi) &= 2\beta_2 + 6\beta_3 \xi \\ f_2''(\xi) &= 2(\beta_2 - 3\beta_4 \xi) + 6(\beta_3 + \beta_4) \xi \\ &= 2\beta_2 + 6\beta_3 \xi.. \end{aligned}$$

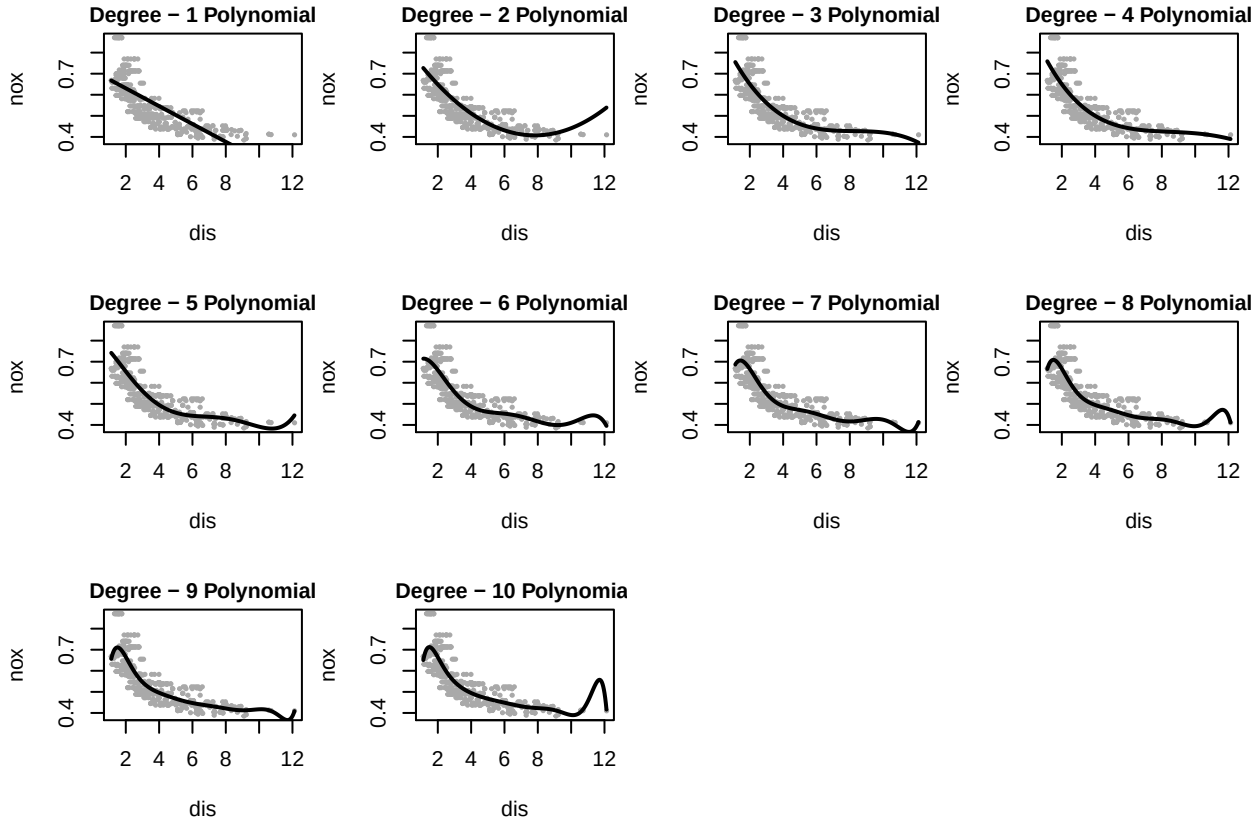
Then, $f_1''(x) = f_2''(x)$ at $x = \xi$, i.e. f"(x) is continuous at ξ.

▶ **Exercises 7.1. Solution.**

(a).



(b). From the plots we can see that all fitted curves are similar within the range of data, while high order curves seem much more wiggle when they come to the out of the bonudary of data. It is no surprising that the highest order polynomial has the lowest residual sum of square since we did not apply any smoothing or regularization method on this approximation.




```
## [1] "RSS for Degree - 1 Polynomial is 2.769"
## [1] "RSS for Degree - 2 Polynomial is 2.035"
```

```
## [1] "RSS for Degree - 3 Polynomial is 1.934"
## [1] "RSS for Degree - 4 Polynomial is 1.933"
## [1] "RSS for Degree - 5 Polynomial is 1.915"
## [1] "RSS for Degree - 6 Polynomial is 1.878"
## [1] "RSS for Degree - 7 Polynomial is 1.849"
## [1] "RSS for Degree - 8 Polynomial is 1.836"
## [1] "RSS for Degree - 9 Polynomial is 1.833"
## [1] "RSS for Degree - 10 Polynomial is 1.832"
```

(c). The 10-fold cross-validation error for each degree is

```
cv.error
```

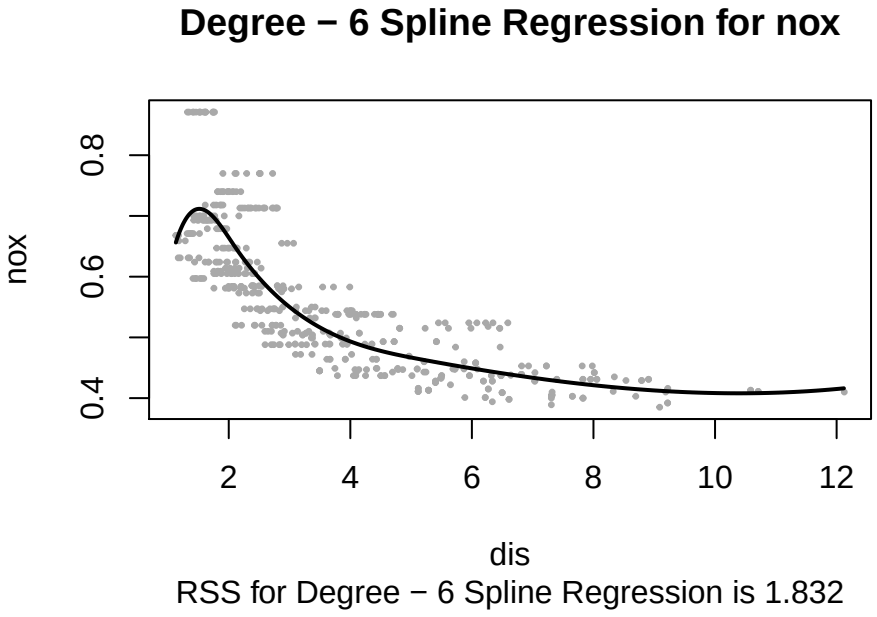
```
##  Degree - 1  Degree - 2  Degree - 3  Degree - 4  Degree - 5  Degree - 6
## 0.009491648 0.030264200 0.009465320 0.023763427 0.025487441 0.004264861
##  Degree - 7  Degree - 8  Degree - 9 Degree - 10
## 0.010033157 0.013344953 0.006236745 0.004148996
```

So the best mode is the Degree - 10 Polynomial Regression with CV error 0.004149. It is a little that the Degree - 10 Polynomial Regression is still not overfitting based on cross-validation. Actually, this result highly depends on the seed of random number. The best model above is selected based on seed=1, but if we change the seed to 5, then the best model will be Degree - 5 Polynomial Regression. One way to solve this problem is to consider the model based on one standard rule as a better model since it is usually more stable.

Another way is to perform this cross-validation multiple times with different, then we will choose the model which is most probablyt to have the least CV mse. I try from seed 1 to seed 100, and the results shows that Degree - 9 Polynomial Regression should be the best model.

(d).

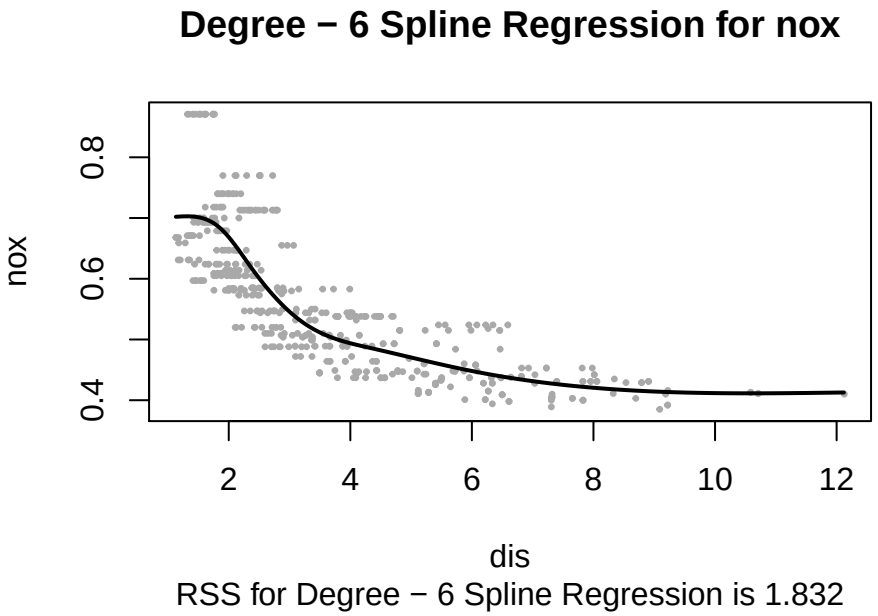
```
## The best model is:  Degree - 6 Spline Regression ;
##  associated degree of freedom:  6 ;
##  associated RSS:  0.1863312 .
##  The plot is shown below
```



(e).

```
## The best model is:  Degree - 6 Spline Regression ;
##  associated degree of freedom:  6 ;
```

```
##  associated RSS:  0.1863312 .
##  The plot is shown below
```



(f). Since the funtion "smooth.spline" can automatically optimize the smoothing parameter by setting the option "cv = TRUE", so we can directly get the smoothing level from the value of "lambda" in object of "smooth.spline" class. Thus, the smoothing level of smooth spline model is

```
ss.fit$lambda
```

```
## [1] 0.07031691
```

(g). Fitting the loess and perform cross-validation should be more careful, because there must be one training sample which does not contain the largest observation, and also one training sample which does not contain the smallest observation (it will very possible that they are in the same test sample), then we will get an error about data out of boundary. I just ignore this single abnormal observation by setting "na.rm = TRUE". Then I still use 10-fold cross validation to choose the best span and the result is

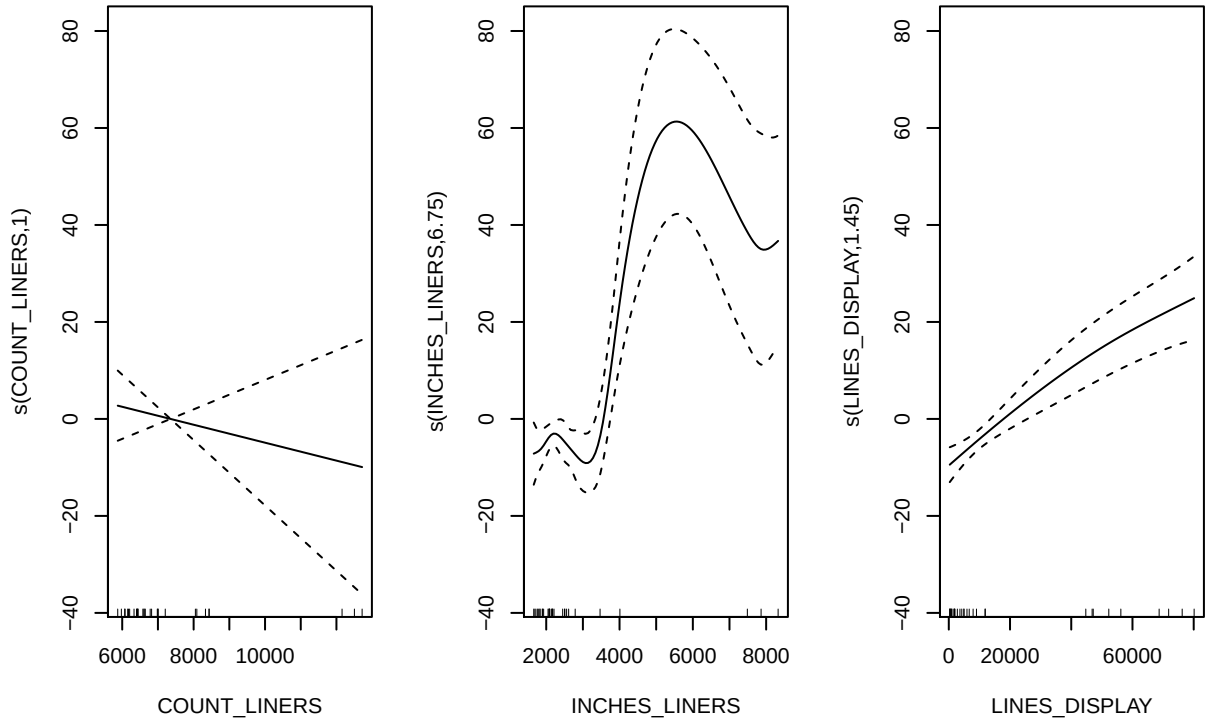
```
cat("The best span is: ",
    best.span,
    "associated MSE: ",
    least.loess.cv)
```

```
## The best span is:  0.35 associated MSE:  0.003715029
```

► **Exercises 3. Solution.**

For this problem, I actually have two way to select the good GAM. For we can fit the GAM with all predictor, i.e.

$$PAGES = \beta_0 + \beta_1 f_1(COUNT\ LINERS) + \beta_1 f_2(INCHES\ LINERS) + \beta_1 f_3(LINES\ DISPLAY)$$


s(COUNT_LINERS,1)
80
60
40
20
0
−20
−40
6000
8000
10000
COUNT_LINERS
s(INCHES_LINERS,6.75)
2000
4000
6000
8000
INCHES_LINERS
s(LINES_DISPLAY,1.45)
0
20000
60000
LINES_DISPLAY