# STAT 8330 FALL 2015 ASSIGNMENT 4

*Peng Shao*

*September 27, 2015*

▶ **Exercises 2. Solution.**

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

(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 $\xi$.

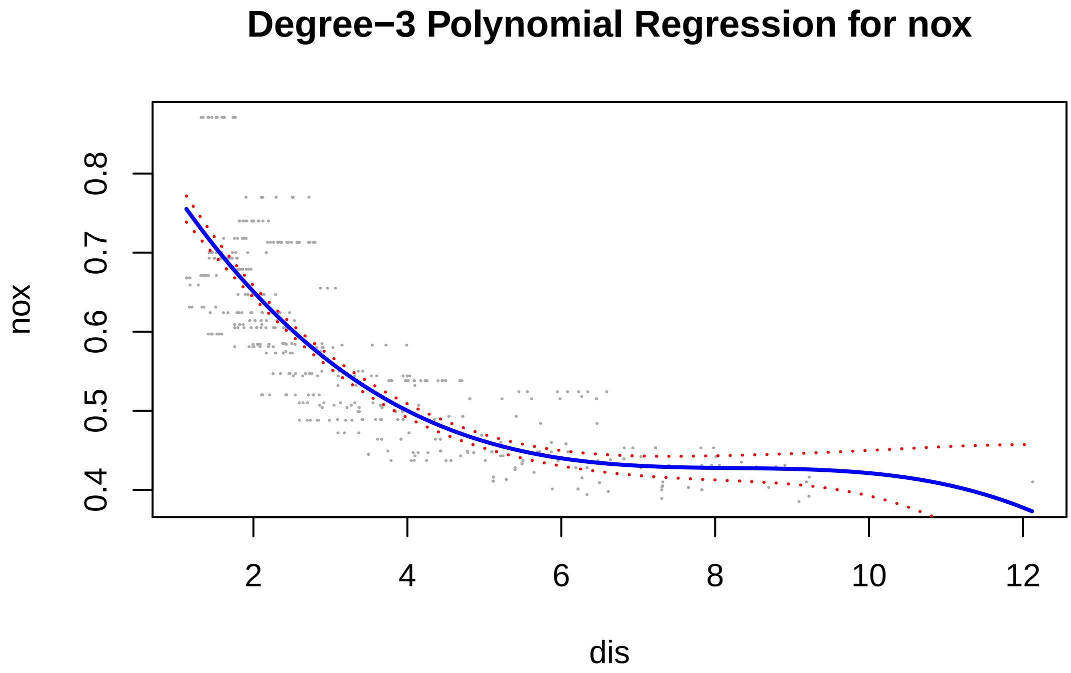
(e).

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

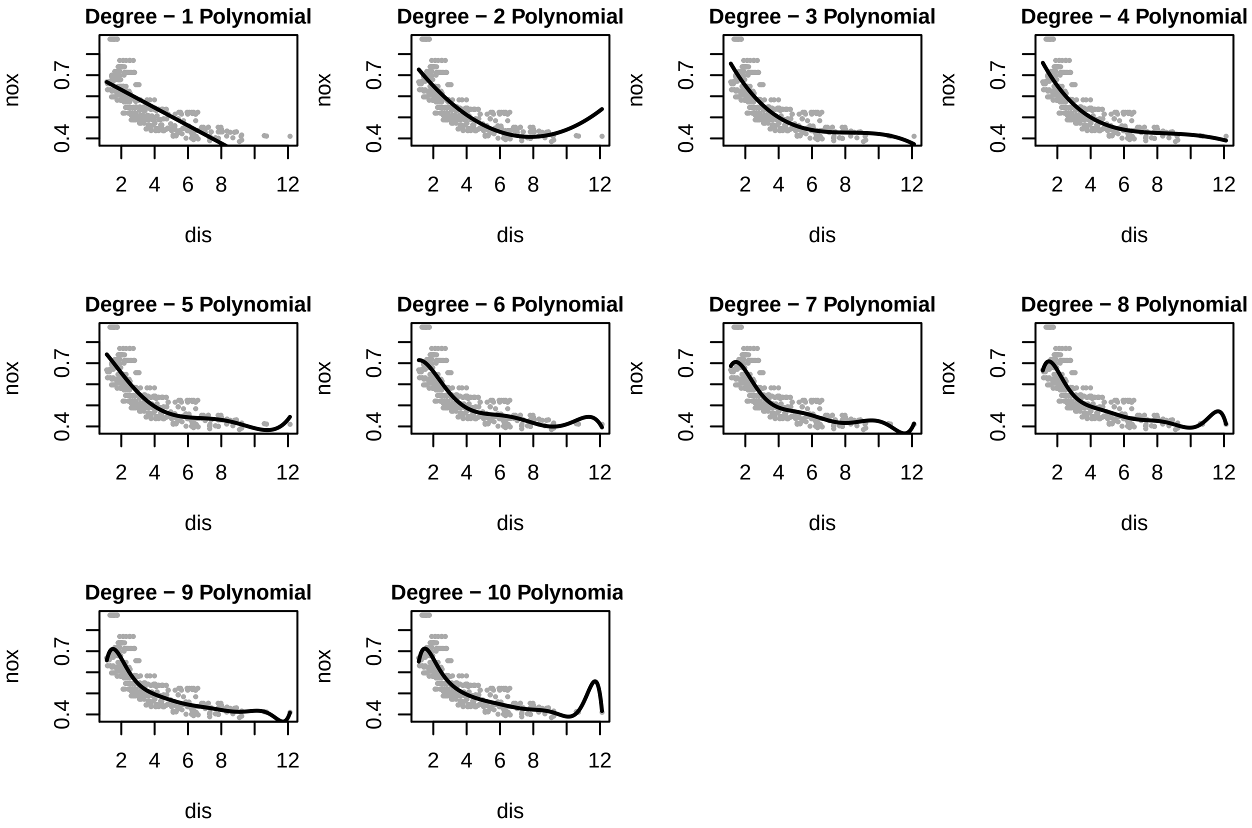
Then, $f_1''(x) = f_2''(x)$ at $x = \xi$, i.e. f"(x) is continuous at $\xi$.

▶ **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
```

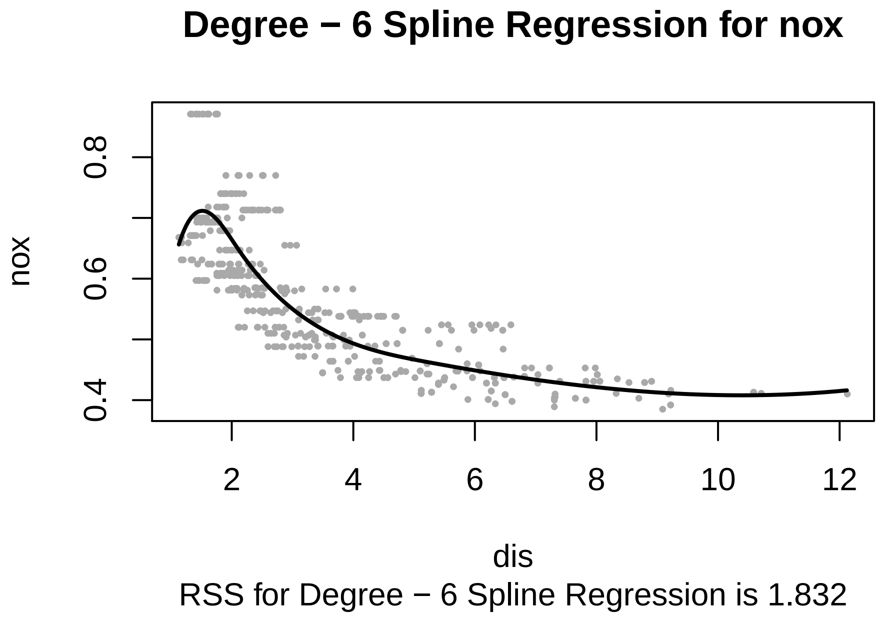
```
##  Degree - 1  Degree - 2  Degree - 3  Degree - 4  Degree - 5  Degree - 6
## 0.009491648 0.030264200 0.009465320 0.023763427 0.025487441 0.004264861
##  Degree - 7  Degree - 8  Degree - 9 Degree - 10
## 0.010033157 0.013344953 0.006236745 0.004148996
```

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.

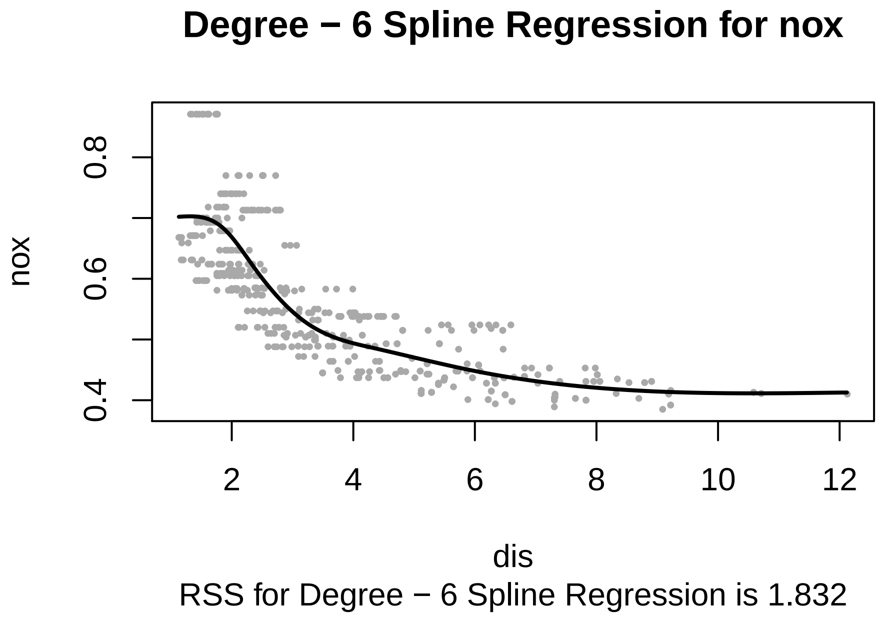
(d).

```
## 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)
```

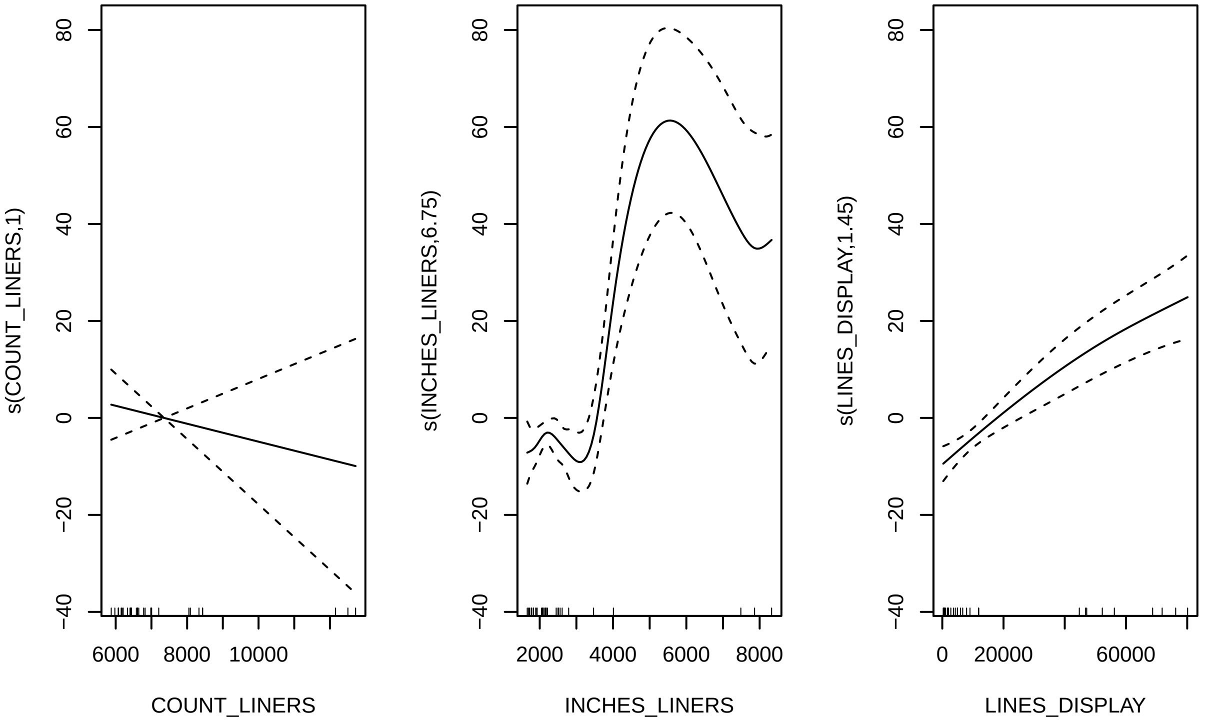
```
## The best span is:  0.35 associated MSE:  0.003715029
```

▶ **Exercises 3. Solution.**

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

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

and get the sequential plots of approximate curves of different predictors. Then we can see that the variable “count linears” is actually completely linear, which efficient degree equals 1, the variable “inches linears” is totally nonlinear with very high variance, which efficient degree equals 6.75, and the variable “lines display” is almost linear with only 1.45 degree, but we don’t want treat it as linear in case of omitting some information. Then we use 5 fold cross-validation for 7 candidate models, which each predictor will be included or not, but at least one in. Finally we get the “best” model stated as below.



```
## The BEST model is:  PAGES ~ s(LINES_DISPLAY) ; associating with error:  69.11991
```

The other way is that we will treat all of them as smooth spline, instead of considering the linearity of each variable, since straight line is just a special case of smooth spline which has very large smooth parameter. Again, we use 5 fold cross-validation for 7 candidate models, the result is

```
## The BEST model is:  PAGES ~ s(LINES_DISPLAY) ; associating with error:  69.11991
```

The results are exactly same, which indicates that these two way is essentially identical.

Furthermore, to compare with the linear model, maybe we can perform some anova test,

```
anova(linear.gam.fit, best.gam.fit.1, best.gam.fit.2, test="F")
```

```
## Analysis of Deviance Table
##
## Model 1: PAGES ~ COUNT_LINERS + INCHES_LINERS + LINES_DISPLAY
## Model 2: PAGES ~ s(LINES_DISPLAY)
## Model 3: PAGES ~ s(LINES_DISPLAY)
##   Resid. Df Resid. Dev     Df Deviance      F   Pr(>F)
## 1    27.000     729.07
## 2    21.414     280.48 5.5863   448.59 6.1306 0.000872 ***
## 3    21.414     280.48 0.0000     0.00
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

That is to say, the GAM model is significantly more useful compared to multiple linear regression.

▶ **Exercises 4. Solution.**

This problem is kind of a scientific problem, so the prediction is not the only thing we want get, but also the inferences and interpretations of predictors. So the usage of GAM seems to be very reasonable, because we can make inferences based on the structure of GAM. I firstly perform some simple linear regressions for the response on each variable, and get the p-value of each regressions, which should be the biggest significance for every variable.

```
p.values
```

```
##              sediments             borrow pit                meander
##             0.37761376             0.07843585             0.07577306
##          channel width         floodway width X.constriction.factor.
##             0.08968309             0.25926211             0.04576422
##             land cover              veg width              sinuosity
##             0.98042299             0.07031942             0.03867586
##               dredging             revetement
##             0.04576422             0.98131850
```

Based on the p-values, I keep some predictors –X.constriction.factor., sinuosity, dredging – for training. Like in problem 3, we still use cross validation to select the best model of 7 candidate models. The difference is that the cost function we need to minimize here is the error. So, the best model should have the smallest validation error rate.

```
gam.formula
```

```
## Failure1 ~ s(sinuosity)
```

```
cvmse[i]
```

```
## [1] 0.3988235
```

It is a little surprising, since the best model only contains one variables, furthermore, we did not imagine that the best predictor is sinuosity among there candidate in our common sense. But the performance of this model is still not so good since it has almost 40% error rate.

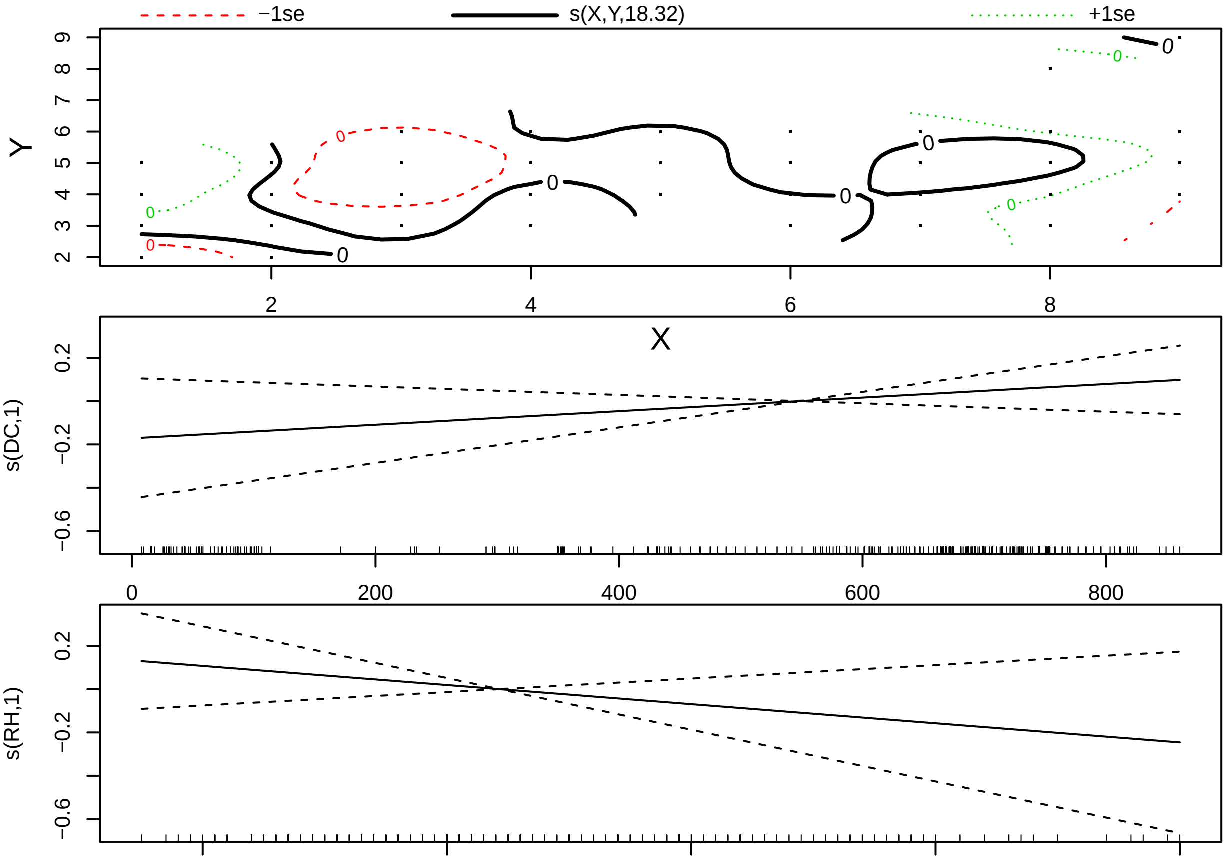
▶ **Exercises 4. Solution.**

The best model I got is "log_area ~ s(X,Y) + s(DC) + s(RH)", and the smoothing parameters is

```
gam.fit$sp
```

```
##       s(X,Y)        s(DC)        s(RH)
## 2.772614e-02 1.932444e+09 1.438130e+09
```

The considerable smoothing parameters for DC and RH indicates that they are linear. But the location variable have very large degree, which more than 18. We verify these from the plots.



The CV is

```
summary(gam.fit)$sp.criterion
```

```
## [1] 1.956103
```

The best model in HW3 is the ridge model with regularization parameter $\lambda = 18.9947174$, associated with $MSE_{CV} = 1.9703114$.