Question 1

a) $RSS(\beta_0, \beta_1) = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$

① Take the derivative of RSS and set it to zero with respect to β_0

$$\frac{\alpha}{\alpha \beta_0} = \sum_{i=1}^{n} 2(y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^{n} (y_i - \hat{\beta_0} - \hat{\beta_1} x_i) = 0$$

$$\sum_{i=1}^{n} y_i - n\hat{\beta_0} - \sum_{i=1}^{n} \hat{\beta_1} x_i = 0$$

$$n\hat{\beta_0} = n\bar{y} - n\hat{\beta_1}\bar{x} \quad \text{(all divided by } n\text{)}$$

$$\hat{\beta_0}^{LS} = \bar{y} - \hat{\beta_1}^{LS}\bar{x}$$

② Take the derivative with respect to β_1 and set it to zero

$$\frac{\alpha}{\alpha \beta_1} = \sum_{i=1}^{n} 2x_i (y_i - \beta_0 - \beta_1 x_i) = 0$$

$$\sum_{i=1}^{n} 2x_i y_i - \sum_{i=1}^{n} 2x_i \hat{\beta_0} - \sum_{i=1}^{n} 2x_i \hat{\beta_1} x_i = 0$$

(all divided by 2)

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} \hat{\beta_0} x_i - \hat{\beta_1} \sum_{i=1}^{n} x_i^2 = 0$$

We then replace $\hat{\beta_0}$ by $\bar{y} - \hat{\beta_1}^{LS}\bar{x}$ into the above equation

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} (\bar{y} - \hat{\beta_1}^{LS}\bar{x}) x_i - \hat{\beta_1} \sum_{i=1}^{n} x_i^2 = 0$$

as $\sum_{i=1}^{n} (x_i - \bar{x})^2 = 1$

$$\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} + n\hat{\beta_1}\bar{x}^2 - \hat{\beta_1} \sum_{i=1}^{n} x_i^2 = 0$$

$$\hat{\beta_1}(n\bar{x}^2 - \sum_{i=1}^{n} x_i^2) = n\bar{x}\bar{y} - \sum_{i=1}^{n} x_i y_i$$

$$\hat{\beta_1}^{LS} = \frac{n\bar{x}\bar{y} - \sum_{i=1}^{n} x_i y_i}{n\bar{x}^2 - \sum_{i=1}^{n} x_i^2}$$

$$= -\left(\frac{n\bar{x}\bar{y} - \sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} (\bar{x} - x_i)^2} \right)$$

$= 1$ by premise

$$\hat{\beta_1}^{LS} = \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$$

b) The ridge regression with λ as the penalty for large parameter β has the expression $\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2 + \lambda \sum_{i=1}^{n} \beta_j^2$

We then take derivate respect to β_0 and set it zero

$$\sum_{i=1}^{n} 2(y_i - \hat{\beta_0} - \hat{\beta_1} x_i) + 0 = 0$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n} \hat{\beta_0} - \sum_{i=1}^{n} \hat{\beta_1} x_i = 0$$

$$n\bar{y} - n\beta_0 - n\beta_1\bar{x} = 0 \quad \text{(all divided by } n\text{)}$$

$$\bar{y} - \hat{\beta_0} - \hat{\beta_1}\bar{x} = 0$$

$\hat{\beta}_{0,\lambda}^{R} = \bar{y} - \hat{\beta}_{1,\lambda}^{R}\bar{x}$ Then we take the derivative respect to β_1 and set it to zero

$$\sum_{i=1}^{n} 2x_i(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) + \lambda 2\hat{\beta}_1 = 0$$

$$\sum_{i=1}^{n} 2x_i y_i - \sum_{i=1}^{n} 2\hat{\beta}_0 x_i - \sum_{i=1}^{n} 2x_i\hat{\beta}_1 x_i + \lambda 2\hat{\beta}_1 = 0$$

(all divided by 2 and replace $\hat{\beta}_0 = \bar{y} - \hat{\beta}_{1,\lambda}^{R}\bar{x}$)

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i(\bar{y} - \hat{\beta}_{1,\lambda}^{R}\bar{x}) - n\hat{\beta}_1^{R}\bar{x}^2 + \lambda\hat{\beta}_1^{R} = 0$$

$$\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y} + \sum_{i=1}^{n} x_i\hat{\beta}_1^{R}\bar{x} - n\hat{\beta}_1^{R}\bar{x}^2 + \lambda\hat{\beta}_1^{R} = 0$$

$$\hat{\beta}_1^{R}\left(\sum_{i=1}^{n} x_i^2 - n\bar{x}^2 + \lambda\right) = \hat{\beta}_1^{LS}$$

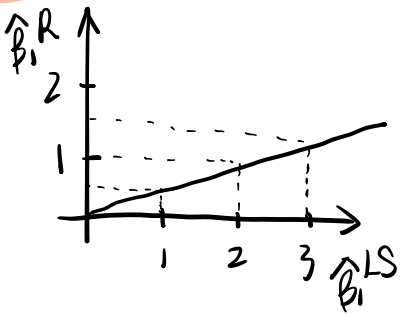
(because $\hat{\beta}_1^{LS} = \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$)

$$\hat{\beta}_1^{R}(1+\lambda) = \hat{\beta}_1^{LS}$$

(because $\sum(x_i - \bar{x})^2 = 1$ as premise)

$$\hat{\beta}_1^{R} = \frac{\hat{\beta}_1^{LS}}{1+\lambda}$$

c) When $\lambda = 1$, $\hat{\beta}_1^{R} = \frac{\hat{\beta}_1^{LS}}{1+1} = \hat{\beta}^{LS}/2$



This is a linear relationship between $\hat{\beta}_1^{LS}$ and $\hat{\beta}_1^{R}$, the ridge regression $\hat{\beta}_{1,\lambda}^{R}$ will not be zero indicating that the coefficient will not be zero for any predictor unless $\hat{\beta}_1^{LS}$ is also zero. Thus, the ridge regression doesn't perform predictor selection.

d) Lasso estimator $(\hat{\beta}_{0,\lambda}^{L}, \hat{\beta}_{1,\lambda}^{L})$

$$\min_{\beta_0,\beta_1}\{RSS(\beta_0,\beta_1) + \lambda|\beta_1|\} \Rightarrow \sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 + \lambda|\beta_1|$$

Take the derivative with respect to β_0 and set it to zero.

$$\sum_{i=1}^{n} 2(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i) + 0 = 0$$

$$\sum_{i=1}^{n} y_i - \sum_{i=1}^{n}\hat{\beta}_0 - \sum_{i=1}^{n}\hat{\beta}_1 x_i = 0$$

$$n\bar{y} - n\hat{\beta}_0 - n\hat{\beta}_1\bar{x} = 0$$

$$\bar{y} - \hat{\beta}_{1,\lambda}^{L}\bar{x} = \hat{\beta}_{0,\lambda}^{L}$$

When $\hat{\beta}_1^{LS} > \frac{\lambda}{2}$, $\hat{\beta}_{1,\lambda}^{L} = \hat{\beta}_1^{LS} - \frac{\lambda}{2}$, for which $\partial\beta_1 = 1$

$$-2\sum_{i=1}^{n} x_i(y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i) + \lambda = 0 \quad \text{(all divided by -2)}$$

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} \hat{\beta}_0 x_i + \sum_{i=1}^{n} \hat{\beta}_1 x_i^2 = \frac{\lambda}{2} \quad \text{(then replace } \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}\text{)}$$

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i(\bar{y} - \hat{\beta}_1 \bar{x}) + \sum_{i=1}^{n} \hat{\beta}_1 x_i^2 = \frac{\lambda}{2}$$

$$\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \bar{y} + \hat{\beta}_1 \bar{x} \sum_{i=1}^{n} x_i + \hat{\beta}_1 \sum_{i=1}^{n} x_i^2 = \frac{\lambda}{2}$$

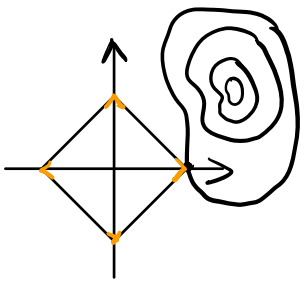
$$\sum_{i=1}^{n} x_i y_i - \bar{y} \sum_{i=1}^{n} x_i + \hat{\beta}_1 \underbrace{\left(\bar{x} \sum_{i=1}^{n} x_i + \sum_{i=1}^{n} x_i^2\right)}_{=1 \text{ as premise}} = \frac{\lambda}{2}$$

as $\hat{\beta}_1^{LS} = \sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}$

$$-\hat{\beta}_1^{LS} + \hat{\beta}_{1,\lambda}^{L} = \frac{\lambda}{2}$$

$$\hat{\beta}_{1,\lambda}^{L} = \hat{\beta}_1^{LS} - \frac{\lambda}{2}$$

$$\hat{\beta}_{0,\lambda}^{L} = \bar{y} - \hat{\beta}_{1,\lambda}^{L} \bar{x}$$



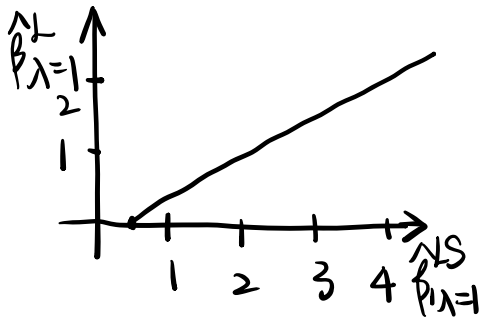
When $\hat{\beta}_1^{LS} \le \frac{\lambda}{2}$, $\hat{\beta}_{1\lambda}^{L} = 0$, then $\partial\beta_1 = [-1, 1]$

$$-2\sum_{i=1}^{n} x_i(y_i - \hat{\beta}_0 + \hat{\beta}_1 x_i) + \lambda \partial\beta_1 = 0.$$

↓ no results applicable

↓ $\hat{\beta}_{0,\lambda}^{L} = \bar{y} - \hat{\beta}_{1,\lambda}^{L} \bar{x}$, $\hat{\beta}_{1\lambda}^{L} = (\hat{\beta}_1^{LS} - \lambda/2)_+$

e) When $\lambda = 1$, $\hat{\beta}_0^{L} = \bar{y} - \hat{\beta}_{1,\lambda}^{L} \bar{x}$, $\hat{\beta}_{1,\lambda=1}^{L} = (\hat{\beta}_1^{LS} - \lambda/2)_+ = \max\{\hat{\beta}_1^{LS} - \frac{1}{2}, 0\}$

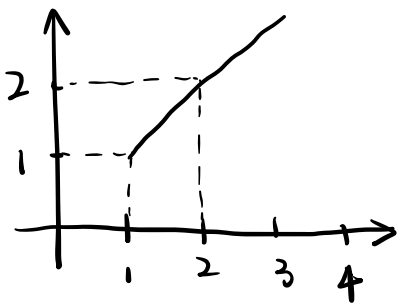


We can see from the lasso, it performs predictor selection by setting the coefficient of $\beta_1 = 0$ when $\hat{\beta}^{L} = 1/2$ (shrinkage the $\hat{\beta}^{LS} = 0$), instead of add all predictors to the final model as ridge did. (predictors with $\hat{\beta}_1^{LS} = 0.5$ are inactive and are zero)

f) minimize $\left\{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_i x_{ij}\right)^2\right\}$ subject to $\sum_{i=1}^{n} I(\beta_j \neq 0) \le s$,

This constraints means that no more than s coefficients can be non-zero, which we have seen before the same as best subset selection because it goes through all $\binom{n}{s}$ models containing s predictors, which is not computationally feasible and convenient as ridge regression or lasso.

h) when $\lambda = 1$, $\hat{\beta}^{S}_{1,\lambda} = \hat{\beta}^{LS}_{1} \cdot I(\hat{\beta}^{LS}_{1} \geq 1)$



It also perform predictor selection because it shrink predictor with $\hat{\beta}^{LS}_{1} < 1$ all to zero and consider them are inactive.

2. [21 points] State whether each of the following statements are TRUE or FALSE. Briefly justify why in a couple of sentences.
(a) Least-squares estimation should be used over ridge regression when there is high multi-collinearity in the data.
False. Least-squares estimation will generate a high-variance and unbiased model when multi-collinearity existed, and ridge regression will perform shrinkage with an optimal lambda value that balance between bias and variance for a smaller MSPE.

(b) Lasso should be used over ridge regression when we know a priori that only a small handful of predictors are active.
True. Ridge regression perform shrinkage but no predictor selection, adding all p predictors to the final model ultimately (\beta will never equal to zero for any \lambda). The Lasso perform shrinkage that penalize larger \beta as well as predictor selection and will select a set of parameters.
smaller

(c) Piecewise polynomial models can be discontinuous without constraints.
True. Piecewise polynomial regression is supposed to have discontinuity at knots in terms of its fitted curves' high flexibility; if we want to reduce the complexity of a piecewise polynomial model, we can add constraints on our piecewise polynomial model that the fitted curve must be continuous to free up the number of degrees of freedom which simplifies the model.

(d) For cubic splines, the variance of the fitted model decreases as more knots are added.
False. As we increase the number of knots, we increase the flexibility of the model; however, we run into a global polynomial fit that overfits the model by giving poor predictions near domain boundaries. Thus, there is a trade-off between variance and bias as we initially have high bias but low variance, and then the bias decrease but the variance increases when we make the model more complex.

(e) Splines provide greater model flexibility in regions with many knots.
True. Increasing the number of knots will increase the flexibility of our model.

(f) A model with high degrees-of-freedom implies a greater bias in its fit.
False. Higher degrees-of-freedom implies more parameters involved into this model and higher flexibility and complexity for which has lower penalty and appears wiggly on approximation. Thus, the bias will actually decrease because the polynomial model will fit closely with each observation or run into a global polynomial that contains high variance but low bias.

(g) Generalized additive models can be much more computationally expensive to fit compared to multiple linear regression.
True. The GAM fits separate non-linear function fj for each predictor xj and allows for additive structure of linear models, for which is much more complicated than fitting only a multiple linear regression. As the GAM contains interpretability for each model based on each predictor and still contains high predictability, it costs more intuitively.

3. [21 points] Consider a quartic spline model with distinct knots ξk, k = 1, · · · , K. A quartic spline satisfies two properties: (i) it is a quartic (i.e., degree-4) polynomial between any two neighboring knots, and (ii) it has continuous derivatives of up to order 3 at each knot. Note that property (ii) includes derivatives of order 0, meaning the quartic spline should be continuous at knots.

a. [5 points] Write out the full model specification for the quartic spline, including model parameters and basis functions (see Equation (7.9) in ISL). How many degrees-of-freedom (d.f.s) are in your model? The full model specification:
$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \beta_4 b_4(x_i) + \ldots + \beta_{K+4} b_{K+4}(x_i) + \epsilon_i$ The basis functions: $b_1(x_i) = x_i$ $b_2(x_i) = x_i^2$ $b_3(x_i) = x_i^3$ $b_4(x_i) = x_i^4$ $b_{K+4}(x_i) = (x - \xi k)_+^4$ for k = 1,...,K, where$(x - \xi k)_+^4 = (x - \xi k)^4$if $x_i >$ $ \xi_k$ and it equals to 0 otherwise. This quartic spline with degree of 4 has K + 5 parameters with K knots(the degrees of freedom is also K+5).

b. To prove (i), it is clear that the model contains quartic characteristics as displayed as above. Between any two consecutive knots, there is a quartic spline model:

Before the first knot : $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \varepsilon_i$

From the first knot K=1 to the very last seuond K-1 : $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \cdots + \beta_{K-1}(x_i - \xi_{K-1})^4 + \varepsilon_i$

After the last knot K: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \cdots + \beta_{K+4}(x_i - \xi_{K+4})^4 + \varepsilon_i$

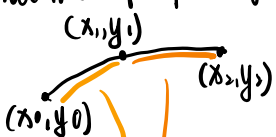
The highest degree of this polynomial model is 4.

We can see that between each two consequtive data points, for example (x_0, y_0) and (x_1, y_1), we can have 2 equations for this quartic model:

$$\begin{cases} \beta_0 + \beta_1 x_0 + \beta_2 x_0^2 + \beta_3 x_0^3 + \cdots + \beta_{K+4}(x_0 - \xi_{K+4})^2 + \varepsilon_i = y_0 \\ \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_1^3 + \cdots + \beta_{K+4}(x_1 - \xi_{K+4})^2 + \varepsilon_i = y_1, \end{cases}$$

For K+4 parameters, we have 2(K+4) equations for solving K+4 unknown β

And the slope for neighboring equations at the same observation is also the same :



Thus, we can take the derivative of both two side and set it to zero

Thus, it is clear that this model contains quartic feature all the way (continuous from the first data point to the very last one)

ii) As above mentioned, we have three points for this quartic spline:

The LIFT: $a_0 + b_0 x_1 + c_0 x_1^2 + d_0 x_1^3 + e_0 x_1^4 = y_1$

The RIGHT: $a_1 + b_1 x_1 + c_1 x_1^2 + d_1 x_1^3 + e_1 x_1^4 = y_1$

> for $x_i > \xi_{K=0}$

LEFT Side :

$y_1' = b_0 + 2c_0 x_1 + 3d_0 x_1^2 + 4e_0 x_1^3$

$y'' = 2c_0 + 6d_0 x_1 + 12 e_0 x_1^2$

$y''' = 6d_0 + 24 e_0 x_1$

$y'''' = 24 e_0$

RIGHT Side:

$y_1' = b_1 + 2c_1 x_1 + 3d_1 x_1^2 + 4e_1 x_1^3$

$y'' = 2c_1 + 6d_1 x_1 + 12 e_1 x_1^2$

$y''' = 3d_1 + 24 e_1 x_1$

$y'''' = 24 e_1$

$\neq 0 \neq$

From the above simple example, we can see that the first, second, and third derivative from each side equal to each other and so that they are continuous. However, the fourth derivative of both sides do not equal to each other and are not zero.

c) No, I don't agree with her.

Because the quartic spline with the formula form of a) the same coefficients estimation, standard errors, confidence intervals, hypothesis testing as the linear regression model, as the quartic spline is transformed to a linear spline with the usage of basis function $b_1, b_2, b_3, \ldots, b_{K+4}$

d) No, I don't agree with her.

The polynomial model with a slightly smaller df of 15 will generate a more flexible and complex model than our quartic spline model when the true regression function changes rapidly in certain regions, but not in others.
The true regression function can have better predictability with low-order polynomial in local regions, and is never a 15-degree polynomial over the full input space.
On the other hand, higher-order polynomial model is unstable and performs poorly at the boundaries where our x takes extremly large or small value, whereas the quartic spline is more stable and have better interpretability.

STA325_HW3_Kedi2

##Question 4

```
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

```
library(ggplot2)
data("Boston")
attach(Boston)
```

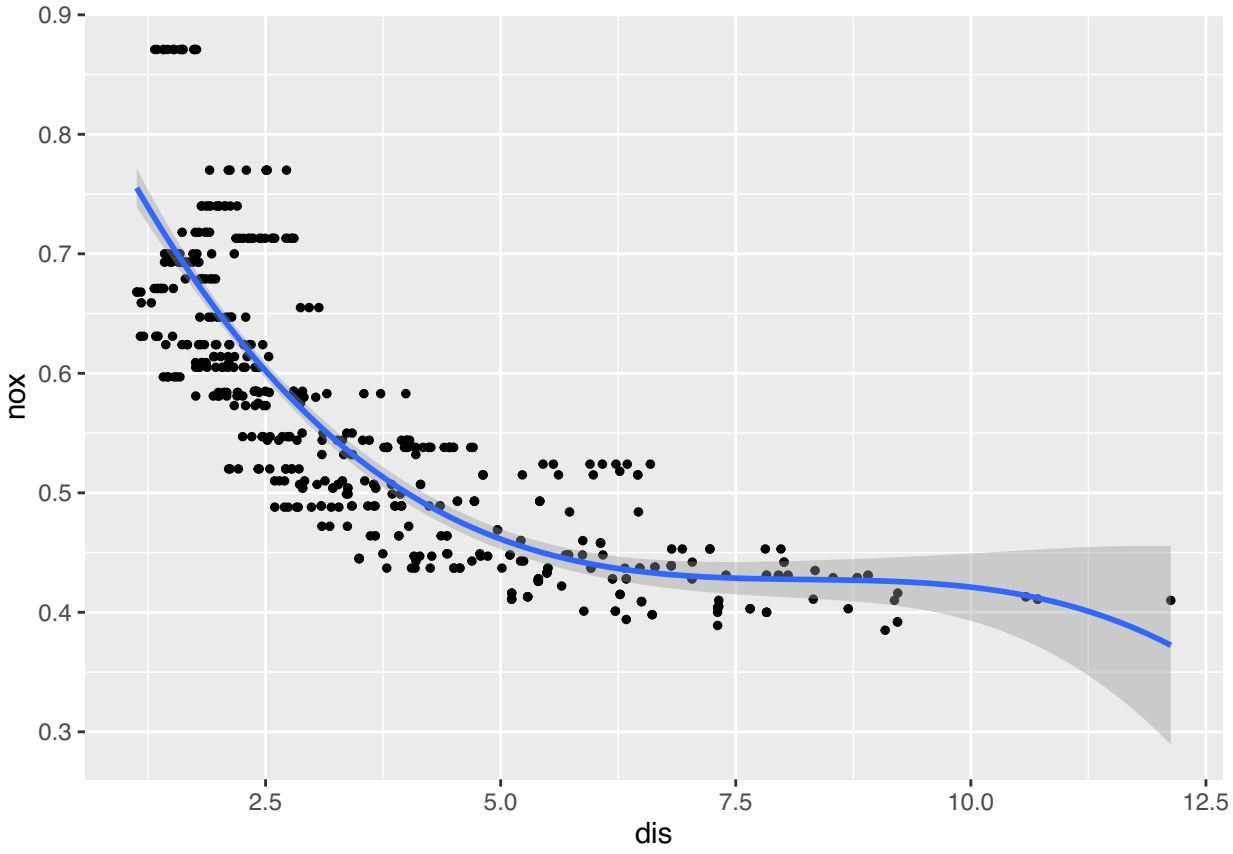
#part a Use the poly() function to fit a cubic polynomial regression to predict nox using dis. Report the regression output, and plot the resulting data and polynomial fits.

```
polynomial_regression1 = lm(nox ~ poly(dis, 3), data = Boston)
summary(polynomial_regression1)
```

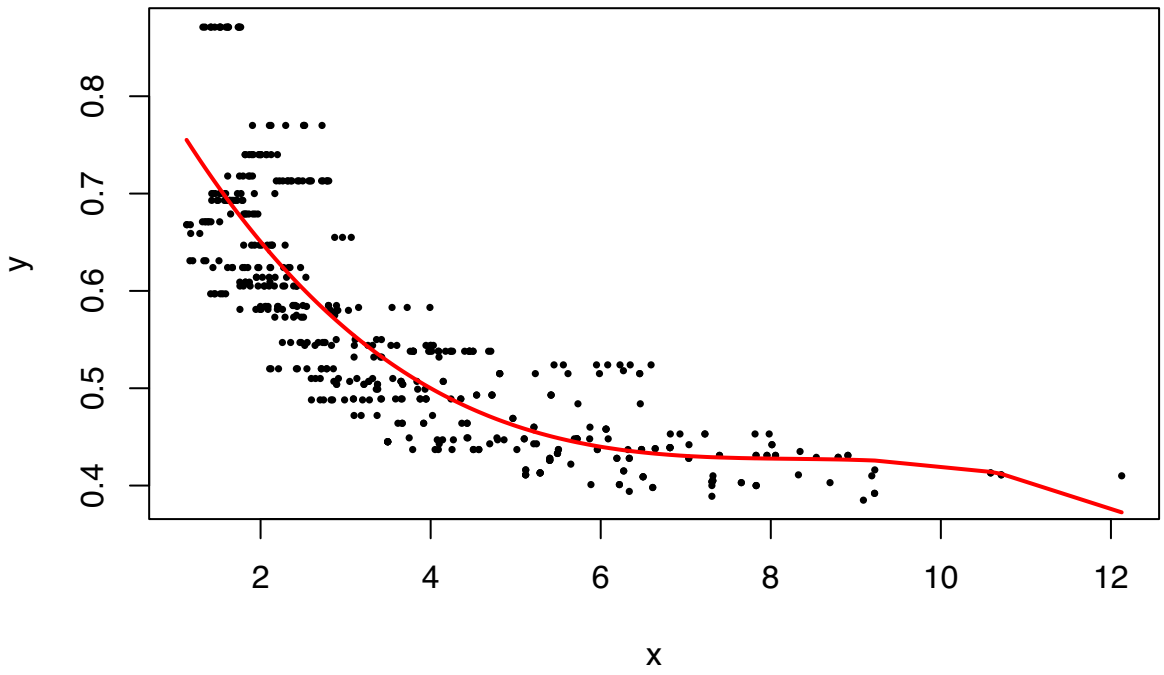
```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 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
```

From the p-value with threshold of $\alpha > 0.05$, we can tell that the fit of a polynomial regression model with degree of 3 is reasonable because all the β are statistically significant.

```
ggplot(Boston, aes(dis,nox)) + geom_point(size=1) + stat_smooth(method = "lm", formula = y ~ poly(x, 3))
```

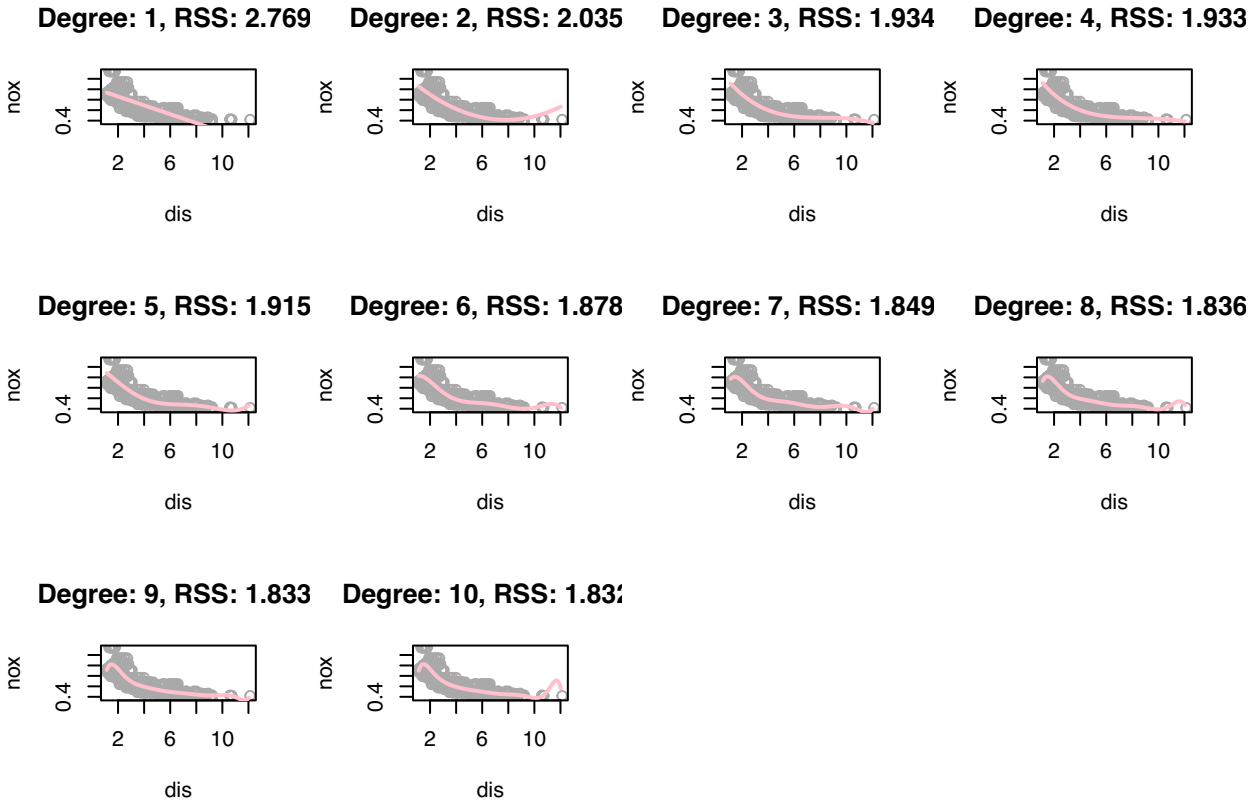



```r
#plot x vs. y
x <- Boston$dis
y <- Boston$nox
plot(x, y, pch=16, cex=0.5)

#fit polynomial regression model
fit <- lm(y ~ x + I(x^2) + I(x^3))

#use model to get predicted values
pred <- predict(fit)
ix <- sort(x, index.return=T)$ix

#add polynomial curve to plot
lines(x[ix], pred[ix], col='red', lwd=2)
```



#part 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.

```
par(mfrow=c(3,4))
rss <- rep(0,10)
for(d in 1:10){
  lm.fit <- lm(nox ~ poly(dis, d), data = Boston)
  dislim <- range(Boston$dis)
  x <- seq(from = dislim[1], to = dislim[2], by = 0.1)
  lm.pred <- predict(lm.fit, data.frame(dis = x))
  plot(nox ~ dis, data = Boston, col = "darkgrey")
  lines(x, lm.pred, col = "pink", lwd = 2)
  rss[d] <- sum(lm.fit$residuals^2)
  title(sprintf("Degree: %s, RSS: %s.",d, round(sum(lm.fit $residuals^2), 3)))
}
par(mfrow=c(1,1))
```




```
rss
```

```
##  [1] 2.768563 2.035262 1.934107 1.932981 1.915290 1.878257 1.849484 1.835630
##  [9] 1.833331 1.832171
```

The RSS decreases with higher-order polynomial model because the RSS is about the testing error and always getting smaller if we fit a more complicated model.

#part c Perform cross-validation or another approach to select the optimal degree for the polynomial, and explain your results.

```
library(boot)
fits <- lapply(1:10, function(d){
  fit <- glm(nox ~ poly(dis, d), data = Boston)
  return(list(fit, cv.glm(Boston, fit, K = 10)$delta[2]))
})
cv.errors <- sapply(1:10, function(i){return(fits[[i]][[2]])})
which.min(cv.errors)
```

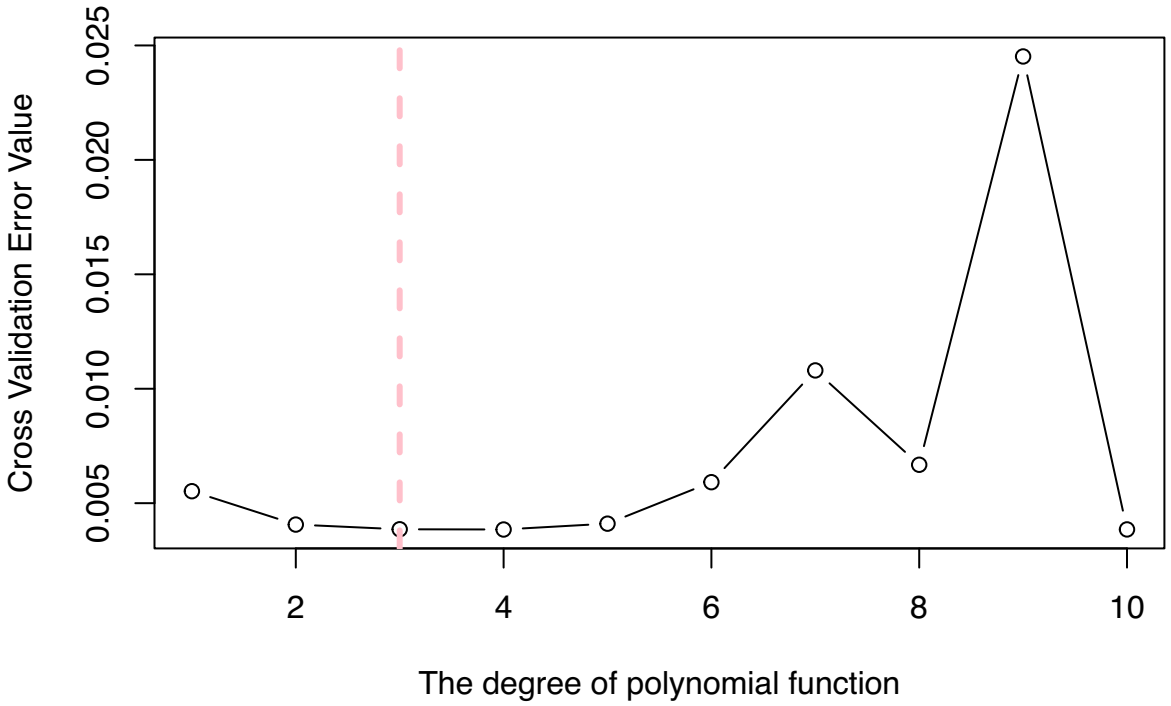
```
## [1] 4
```

```
cv.errors
```

```
##  [1] 0.005523304 0.004066369 0.003860222 0.003850581 0.004105895 0.005917304
##  [7] 0.010801531 0.006679097 0.024518693 0.003853480
```

I use the cross-validation and then lowest error was with a polynomial of fourth degree.

```
plot(cv.errors, type = "b", xlab = "The degree of polynomial function", ylab = "Cross Validation Error "
abline(v = 3, col = "pink", lwd=3, lty=2)
```



If we draw a graph that displays the cross-validation error as a function of the degree of polynomial function, and we can see that the lowest value of error is about at degree of three. So that a cubic polynomial function performs the best among other degrees-of-polynomial function.

#part 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.

The predictor dis's range is from 1.1 to 13.1, and thus we can split it into four intervals and set up knots around at about 4, 7, and 11.

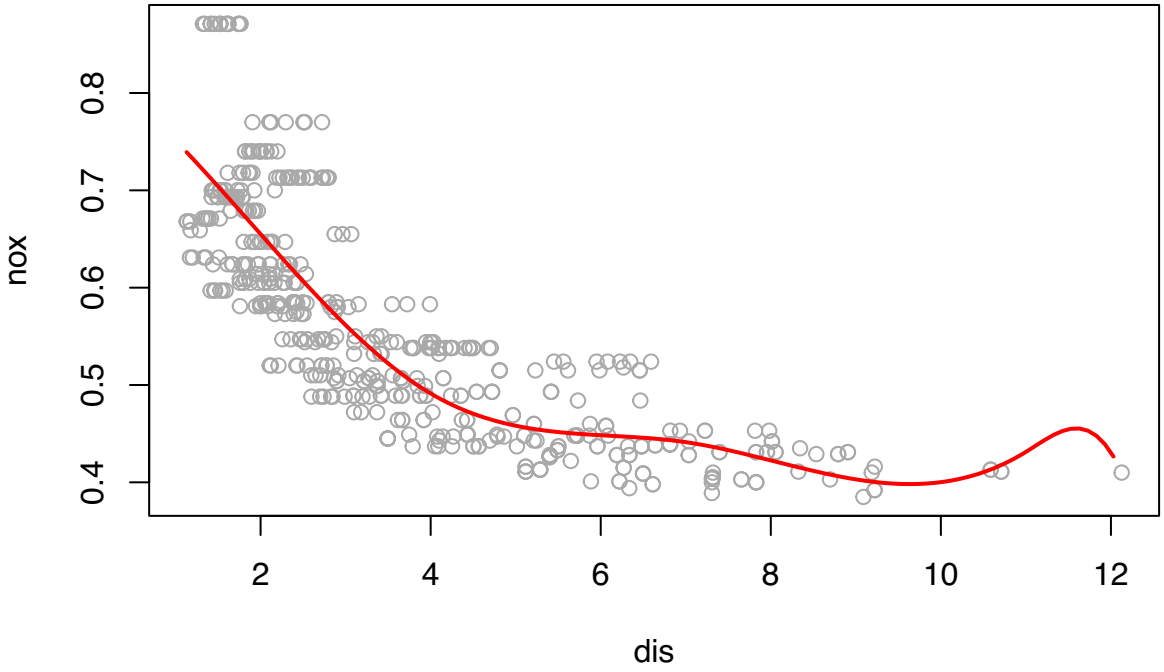
```
maxMin <- range(Boston$dis)
```

```
library(splines)
spline_fit <- lm(nox ~ bs(dis, df = 4, knots = c(4, 7, 11)), data = Boston)
summary(spline_fit)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, df = 4, knots = c(4, 7, 11)), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124567 -0.040355 -0.008702  0.024740  0.192920
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            0.73926    0.01331  55.537  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))1 -0.08861    0.02504  -3.539  0.00044 ***
## bs(dis, df = 4, knots = c(4, 7, 11))2 -0.31341    0.01680 -18.658  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))3 -0.26618    0.03147  -8.459 3.00e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))4 -0.39802    0.04647  -8.565  < 2e-16 ***
## bs(dis, df = 4, knots = c(4, 7, 11))5 -0.25681    0.09001  -2.853  0.00451 **
## bs(dis, df = 4, knots = c(4, 7, 11))6 -0.32926    0.06327  -5.204 2.85e-07 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06185 on 499 degrees of freedom
## Multiple R-squared:  0.7185, Adjusted R-squared:  0.7151
## F-statistic: 212.3 on 6 and 499 DF,  p-value: < 2.2e-16
```

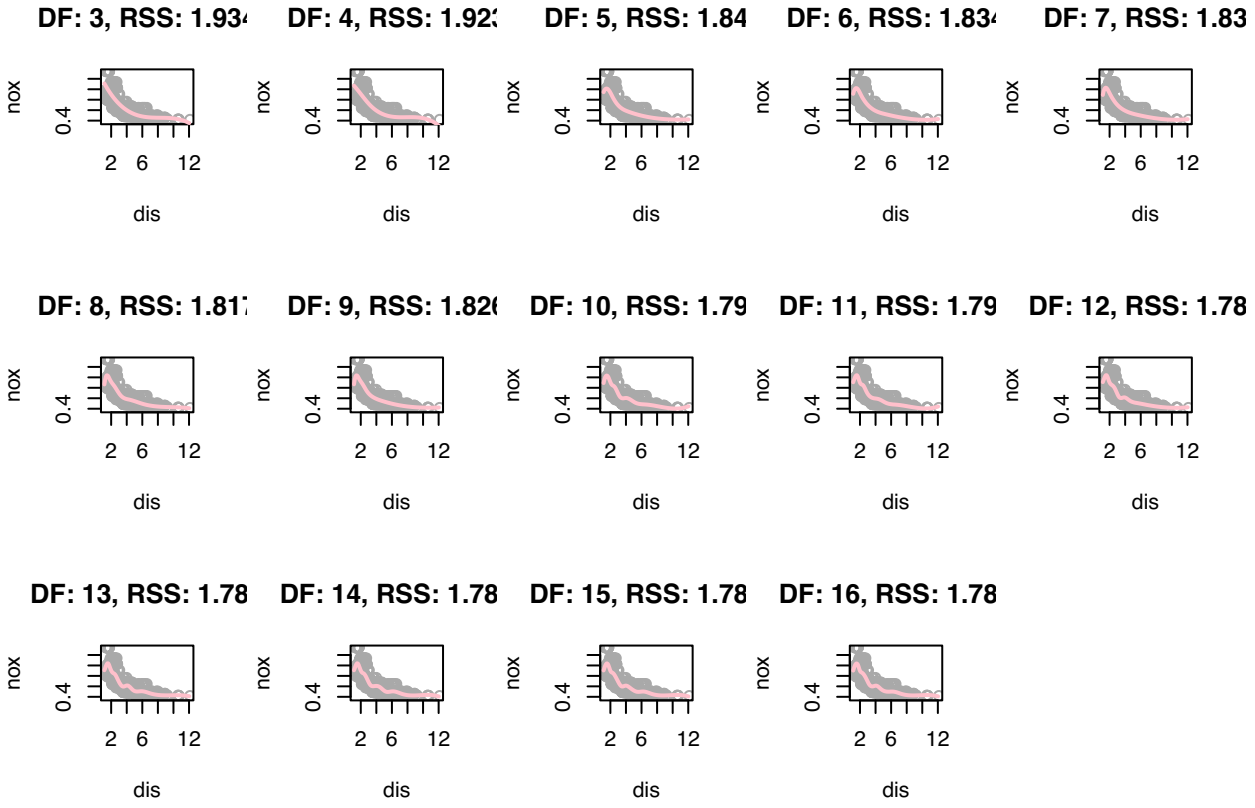
```
prediction_1 <- predict(spline_fit, data.frame(dis = x))
plot(nox ~ dis, data = Boston, col = "darkgrey")
lines(x, prediction_1, col = "red", lwd = 2)
```



A cubic spline with K knots would have K+4 parameters or degrees of freedom. Therefore, with four degrees of freedom we can only have a cubic spline with zero knot. Also, as the above graph displayed, this cubic polynomial function performs well except at the boundary at about x takes value bigger than 10.

#part 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(0, 16)
par(mfrow=c(3,5))
for (i in 3:16) {
    lm.fit = lm(nox ~ bs(dis, df = i), data = Boston)
    rss[i] = sum(lm.fit$residuals^2)
    pred <- predict(lm.fit, data.frame(dis=x))
    plot(nox ~ dis, data = Boston, col = "darkgrey")
    lines(x, pred, col = "pink", lwd = 2)
    title(sprintf("DF: %s, RSS: %s", i, round(sum(lm.fit$residuals^2), 3)))
}
par(mfrow=c(1,1))
```




```
rss[-c(1, 2)]
```

```
##  [1] 1.934107 1.922775 1.840173 1.833966 1.829884 1.816995 1.825653 1.792535
##  [9] 1.796992 1.788999 1.782350 1.781838 1.782798 1.783546
```

For the degree of freedoms, I choose from d.f = 3 to d.f = 16, and we can also see that cubic spline model with a bigger value of degree of freedom will tend to have a smaller RSS, as the model become more complex, flexible, and wiggly. Thus, a model with more degrees of freedom will tend to have higher variance and lower bias, and model with smaller degrees of freedom will have a smaller variance but higher bias, which is the trade off between variance and bias.

#part e

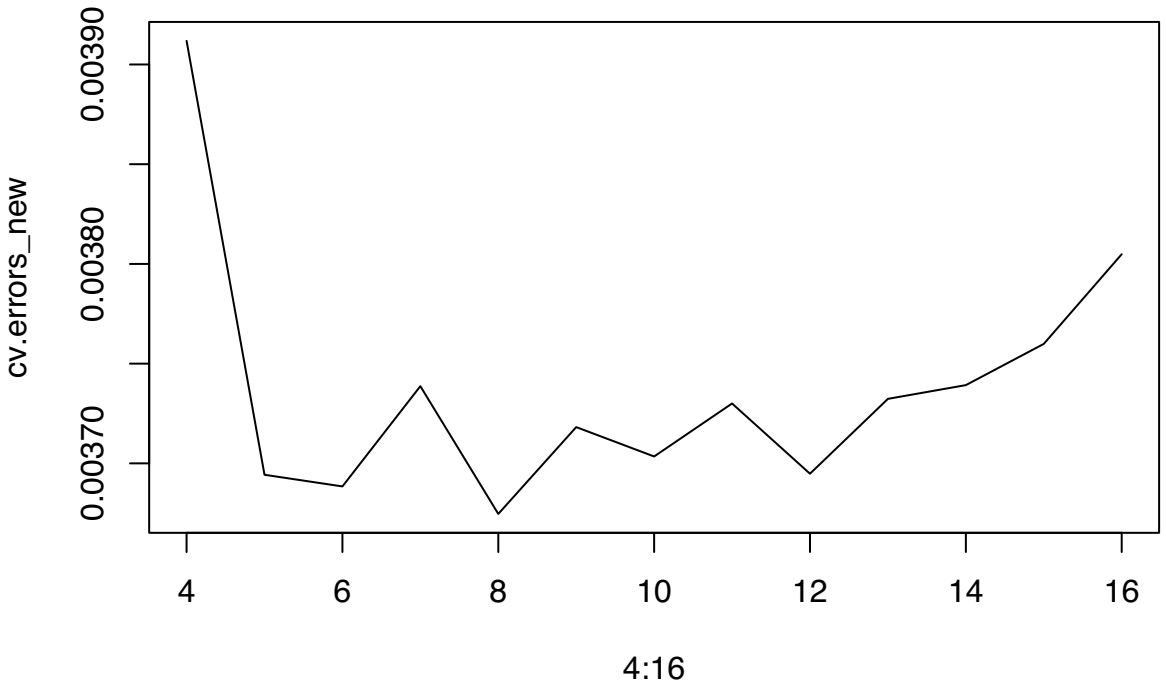
Perform cross-validation or another approach in order to select the best degrees of freedom for a regression spline on this data. Describe your results.

```
cv.errors_new <- sapply(4:16, function(i){
  lm.fit <- glm(nox ~ bs(dis, df = i), data=Boston)
  return(cv.glm(Boston, lm.fit, K = 10)$delta[2])})
```

```
## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.23925), Boundary.knots =
## c(1.137, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.23925), Boundary.knots =
## c(1.137, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.2797), Boundary.knots =
## c(1.1296, : some 'x' values beyond boundary knots may cause ill-conditioned
## bases

## Warning in bs(dis, degree = 3L, knots = c(`50%` = 3.2797), Boundary.knots =
```

```
## 1.8172, : some 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.52943846153846, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.692308%` = 1.52943846153846, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.5284, `14.28571%`
## = 1.80062857142857, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.5284, `14.28571%`
## = 1.80062857142857, : some 'x' values beyond boundary knots may cause ill-
## conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.54201428571429, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases

## Warning in bs(dis, degree = 3L, knots = c(`7.142857%` = 1.54201428571429, : some
## 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
plot(4:16, cv.errors_new,type = "l")
```



It is clear that the cross validation with k = 10 generate a cubic spline with knots of 8 that results the minimum validation error. We choose knots = 8 instead of 8 or 10 or 12 which all contain a relatively small error close to that of 8 because we want to reduce the complexity of the model for its interpretability and avoid the problem of overfitting into a global spline model with high variance.