STAT*6801 Assignment 2

Due: Monday, October 21, 2024 at 11:59pm
(Extended to Wednesday, October 23, 2024 at 11:59pm)

1. Show that the ridge regression estimate is the mean (and mode) of the posterior distribution, under Gaussian prior $\beta \sim N(0, \tau^2\mathbf{I})$, and Gaussian sampling model $y \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I})$. Find the relationship between the regularization parameter λ in the ridge formula, and the variances τ^2 and σ^2.

2. This question pertains to a hypothetical data set arising from an experiment in rats ranging in weight from 100 to 700 grams (average weight is 400 grams). Rats were randomized to one of ten treatment groups receiving varying levels of beta carotene (ranging from 0 to 25 mg) for six weeks. At the end of the experiment, the concentration of vitamin E in the rats' body fat was measured, and the range of values was 130-200 mg/g of body fat. The question of interest is whether the dose of beta carotene affected the concentration of beta carotene in body fat. Let

$$\begin{aligned} Wt &= \text{weight of the rat, and} \\ Dose &= \text{dose of the beta carotene.} \end{aligned}$$

Consider the following models:

Model 2A: $E[VitEconc] = \beta_0 + \beta_1 \times Dose$
Model 2B: $E[log(VitEconc)] = \gamma_0 + \gamma_1 \times Dose$
Model 2C: $log(E[VitEconc]) = \delta_0 + \delta_1 \times Dose$

   (a) How do the interpretations of β_1, γ_1 and δ_1 differ?

   (b) Can linear regression be used to fit each of these models? If so, explain how and provide interpretations of the model parameters in non-statistical terms. If not, explain why not.

3. Consider the truncated power series representation for a cubic spline with K interior knots $\{\xi_1, ..., \xi_K\}$,

$$f(X) = \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k \left(X - \xi_k\right)_+^3$$

   (a) Show that the truncated power series representation satisfies the three conditions (constraints) of a cubic spline.

   (b) (Q5.4 of ESLII, pg 183). Show that the natural boundary conditions for natural cubic splines imply the following linear constraints on the coefficients:

$$\beta_2 = 0, \quad \sum_{k=1}^{K} \theta_k = 0, \beta_3 = 0, \quad \sum_{k=1}^{K} \xi_k \theta_k = 0.$$

4. (Q7.9 of ISLR, pg 324). 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, available in the `ISLR2` package in `R`. 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 tt 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.

1. Show that the ridge regression estimate is the mean (and mode) of the posterior distribution, under Gaussian prior $\beta \sim N(0, \tau^2\mathbf{I})$, and Gaussian sampling model $y \sim N(\mathbf{X}\beta, \sigma^2\mathbf{I})$. Find the relationship between the regularization parameter λ in the ridge formula, and the variances τ^2 and σ^2.

Prior: $p(\beta) \sim N(0, \tau^2 I)$

likelihood: $p(y|X,\beta) \sim N(X\beta, \sigma^2 I)$

Posterior $\propto$ Prior · likelihood

$p(\beta|y) \propto N(0, \tau^2 I) \cdot N(X\beta, \sigma^2 I)$

Posterior $\Sigma_{post} = \left((\tau^2 I)^{-1} + X'(\sigma^2 I)^{-1} X\right)^{-1}$

$\mu_{post} = \Sigma_{post} \cdot \left((\tau^2 I)^{-1} \cdot 0 + X'(\sigma^2 I)^{-1} \cdot y\right)$

$= \left((\tau^2 I)^{-1} + X'(\sigma^2 I)^{-1} X\right)^{-1} \cdot X'(\sigma^2 I)^{-1} \cdot y$

$A^{-1} \cdot B^{-1} = (BA)^{-1} = \left(\sigma^2 I \cdot (\tau^2 I)^{-1} + \sigma^2 I \cdot X'(\sigma^2 I)^{-1} X\right)^{-1} \cdot X'y$

$= \left((\sigma^2 \cdot \tau^{-2}) \cdot I + \sigma^2 I \cdot X'(\sigma^2 I)^{-1} X\right)^{-1} \cdot X'y$

$= \left((\sigma^2 \cdot \tau^{-2}) \cdot I + X'X\right)^{-1} \cdot X'y$

As $\hat{\beta}_{ridge} = (X'X + \lambda I)^{-1} \cdot X'y$

$\Rightarrow \lambda = \sigma^2 \cdot \tau^{-2} = \frac{\sigma^2}{\tau^2}$

2. This question pertains to a hypothetical data set arising from an experiment in rats ranging in weight from 100 to 700 grams (average weight is 400 grams). Rats were randomized to one of ten treatment groups receiving varying levels of beta carotene (ranging from 0 to 25 mg) for six weeks. At the end of the experiment, the concentration of vitamin E in the rats' body fat was measured, and the range of values was 130-200 mg/g of body fat. The question of interest is whether the dose of beta carotene affected the concentration of beta carotene in body fat. Let

$$\begin{aligned} Wt &= \text{weight of the rat, and} \\ Dose &= \text{dose of the beta carotene.} \end{aligned}$$

Consider the following models:

$$\begin{aligned} &\text{Model 2A: } E[VitEconc] = \beta_0 + \beta_1 \times Dose \\ &\text{Model 2B: } E[log(VitEconc)] = \gamma_0 + \gamma_1 \times Dose \\ &\text{Model 2C: } log(E[VitEconc]) = \delta_0 + \delta_1 \times Dose \end{aligned}$$

$E[\text{VitEconc}] = e^{\delta_0} \cdot (e^{\delta_1})^{\text{dose}}$.

(a) How do the interpretations of β_1, γ_1 and δ_1 differ?

β_1 is the unit of change of expected value of VitEconc

when Dose change by 1unit

Γ_1 is the unit of change of expected value of log(VitEconc)

when Dose change by 1unit.

δ_1 is the unit of change of log of VitEconc when Dose change by 1unit.

e^{δ_1} is the unit of change multiplied when dose change by 1 unit.

(b) Can linear regression be used to fit each of these models? If so, explain how and provide interpretations of the model parameters in non-statistical terms. If not, explain why not.

For model 2A. Yes there is obvious linear relationship as

$E[\text{VitEconc}] = \beta_0 + \beta_1 \times \text{Dose}$

β_0 is the basic level of concentration of VitE

and β_1 is the unit effect of Dose on level of VitEConc.

for model 2B. $E[\log(\text{VitZconc})] = \Gamma_0 + \Gamma_1 \cdot \text{Dose}$

$\log(\text{VitZconc}) = \Gamma_0 + \Gamma_1 \cdot \text{Dose} + \varepsilon$

This is linear as Γ_1 is the unit change of $\log(\text{VitZconc})$ when dose increase a level of unit.

Γ_0 is the expected value of $\log(\text{VitZconc})$ with 0 level of Dose.

Thus linear regression can be used on $\log(\text{VitZconc})$ and Dose.

for model 2C.

$\log(E(\text{VitZconc})) = \beta_0 + \beta_1 \cdot \text{Dose}$.

$\text{VitZconc} = e^{\beta_0} \cdot e^{\beta_1 \cdot \text{Dose}}$.

$\text{VitZconc} = e^{\beta_0} \cdot e^{\beta_1 \cdot (\text{Dose}+1)} = e^{\beta_0} \cdot e^{\beta_1} \cdot e^{\beta_1 \text{Dose}}$.

Thus, we can not use directly linear regression of Dose and VitZconc, there only exise relationship on

$\widehat{\text{linear}}$

$\log(E(\text{VitZconc}))$ and dose.

3. Consider the truncated power series representation for a cubic spline with K interior knots $\{\xi_1, ..., \xi_K\}$,

$$f(X) = \sum_{j=0}^{3} \beta_j X^j + \sum_{k=1}^{K} \theta_k \left(X - \xi_k\right)_+^3$$

(a) Show that the truncated power series representation satisfies the three conditions (constraints) of a cubic spline.

$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \theta_1 (x-\xi_1)_+^3 + \theta_2 (x-\xi_2)_+^3 + \cdots + \theta_K (x-\xi_K)_+^3$

for knot $t \in \{\xi_1, \cdots \xi_K\}$

$f(\xi_t^+) = f(\xi_t^-)$

$f(\xi_t^+) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \theta_1 (x-\xi_1)_+^3 + \cdots + \theta_{t-1}(x-\xi_{t-1})_+^3 + \theta_t (x-\xi_t)_+^3 + 0$

$f(\xi_t^-) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \theta_1 (x-\xi_1)_+^3 + \cdots + \theta_{t-1}(x-\xi_{t-1})_+^3 + 0$

As $\xi_t - \xi_t = 0 \Rightarrow \theta_t (x-\xi_t)_+^3 = 0 \Rightarrow f(\xi_t^+) = f(\xi_t^-)$

$f'(\xi_t^+) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2 + 3\theta_1 (x-\xi_1)_+^2 + \cdots + 3\theta_{t-1}(x-\xi_{t-1})_+^2 + 3\theta_t (x-\xi_t)_+^2 + 0$

$f'(\xi_t^-) = \beta_1 + 2\beta_2 x + 3\beta_3 x^2 + 3\theta_1 (x-\xi_1)_+^2 + \cdots + 3\theta_{t-1}(x-\xi_{t-1})_+^2 + 0$

As $\xi_t - \xi_t = 0 \Rightarrow 3\theta_t (x-\xi_t)_+^2 = 0 \Rightarrow f'(\xi_t^+) = f'(\xi_t^-)$

$f''(\xi_t^+) = 2\beta_2 + 6\beta_3 x + 6\theta_1 (x-\xi_1)_+ + \cdots 6\theta_{t-1}(x-\xi_{t-1})_+ + 6\theta_t (x-\xi_t)_+ + 0$

$f''(\xi_t^-) = 2\beta_2 + 6\beta_3 x + 6\theta_1 (x-\xi_1)_+ + \cdots 6\theta_{t-1}(x-\xi_{t-1})_+ + 0$

As $\xi_t - \xi_t = 0 \Rightarrow 6\theta_t (x-\xi_t) = 0 \Rightarrow f''(\xi_t^+) = f''(\xi_t^-)$

As continuous, first derivatives and 2-nd order derivatives are proved for each knots. Thus cubic splines.

(b) (Q5.4 of ESLII, pg 183). Show that the natural boundary conditions for natural cubic splines imply the following linear constraints on the coefficients:

$$\beta_2 = 0, \quad \sum_{k=1}^{K} \theta_k = 0, \beta_3 = 0, \quad \sum_{k=1}^{K} \xi_k \theta_k = 0.$$

As natural boundary means linear on two ends.

$\Rightarrow$ for $x \le \xi_1$: $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$

As linear $\Rightarrow$ $\beta_2 = 0$, $\beta_3 = 0$

$\Rightarrow$ for $x \ge \xi_K$: $f(x) = \beta_0 + \beta_1 x + \theta_1 (x-\xi_1)_+^3 + \theta_2 (x-\xi_2)_+^3 + \dots + \theta_K (x-\xi_K)_+^3$

As linear $\Rightarrow$ $\theta_1 (x-\xi_1)^3 + \dots \theta_K (x-\xi_K)^3 = 0$

$f''(x) = 6\theta_1 (x-\xi_1) + 6\theta_2 (x-\xi_2) + \dots + 6\theta_K (x-\xi_K)$

$= 6x \left(\sum_{K=1}^{K} \theta_K\right) - 6 \left(\sum_{K=1}^{K} \theta_K \xi_K\right) = 0$

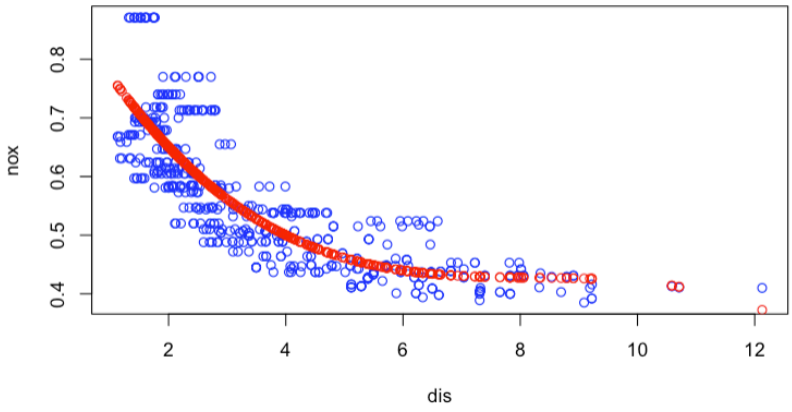
Thus as x is not fixed $\Rightarrow \sum_{K=1}^{K} \theta_K = 0 = \sum_{K=1}^{K} \theta_K \xi_K$

4. (Q7.9 of ISLR, pg 324). 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, available in the `ISLR2` package in `R`. We will treat `dis` as the predictor and `nox` as the response.

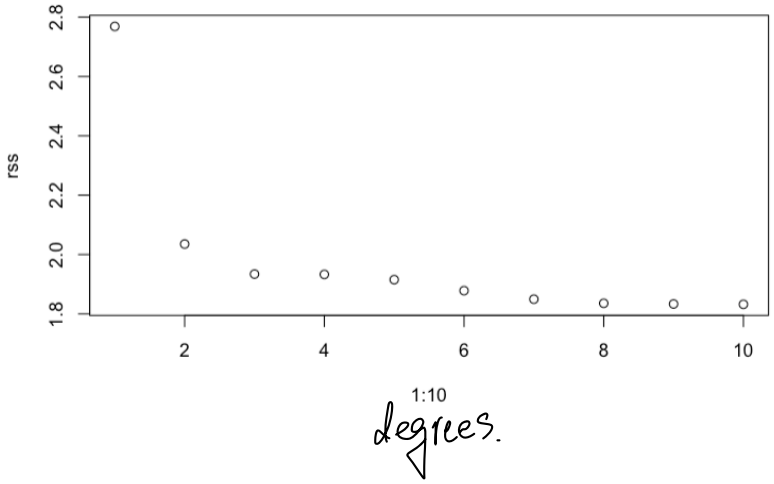
   (a) Use the `poly()` function to fit a cubic polynomial regression to predict `nox` using tt dis. Report the regression output, and plot the resulting data and polynomial fits.

(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.

It's obvious that the training error is decreasing as the degree gets higher, we are approaching over fitting.



a)



degrees.

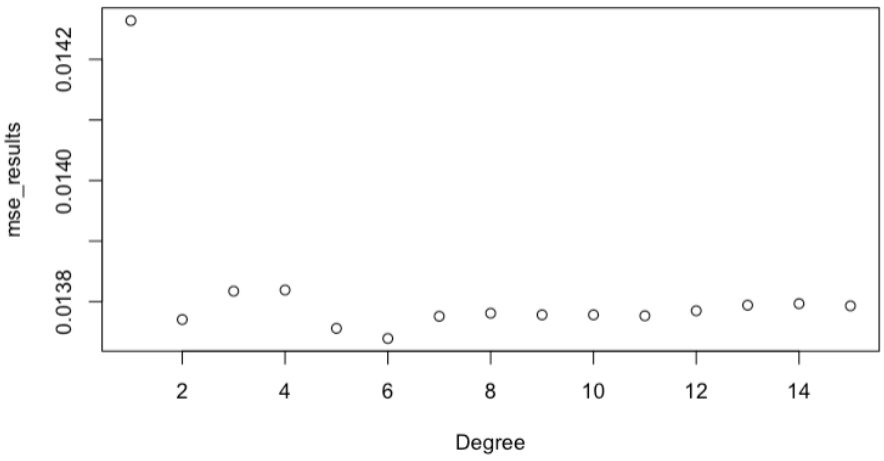
b)

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

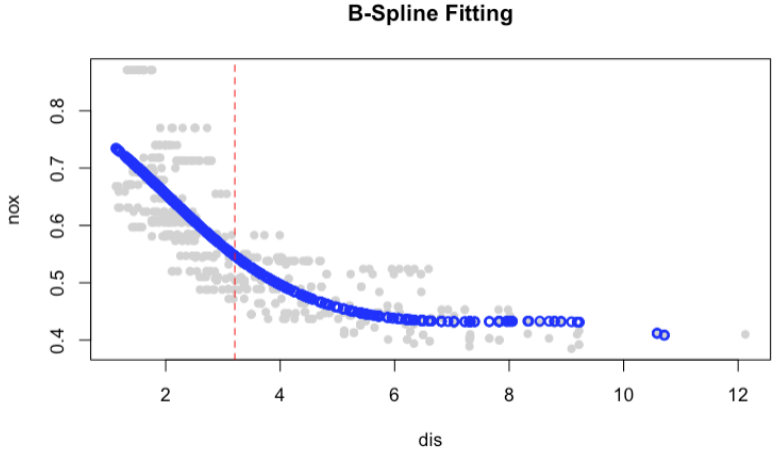
To avoid overfitting, we need to take account of test error. And from the plot, we can tell that the best degree is 6.

(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.

In this case, only one knot is chosen at 3.20745



c)



d)

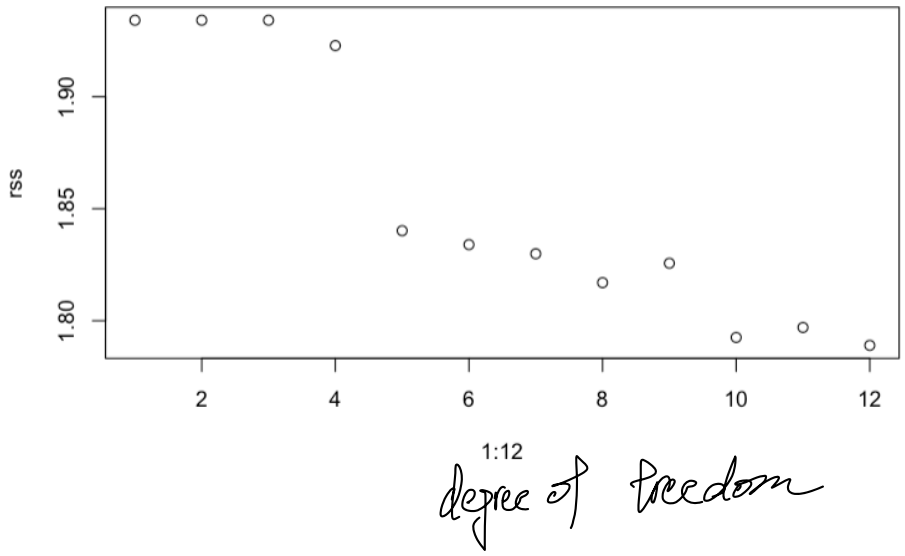
(e) Now fit a regression spline for a range of degrees of freedom, and plot the resulting fits and report the resulting RSS. Describe the results obtained.

As df increases, the more knots are chosen, and the Rss decreases and performance are getting better, which causes the overfitting on the training data.

(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.

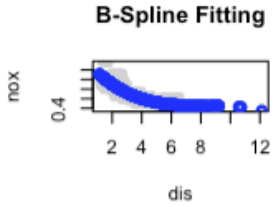
with C-V, it's obvious that test error is lowest at df = 4 and increases after the df gets higher due to higher flexibility which indicates overfitting.

The df = 4 is the balanced trade off.

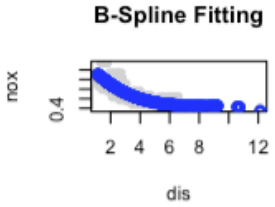


e)

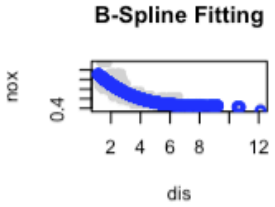
df=3



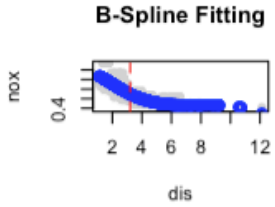
3



3

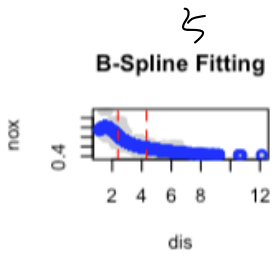


4

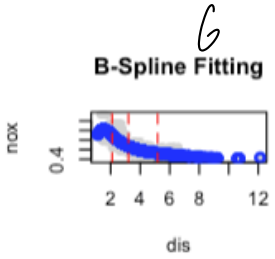


e)

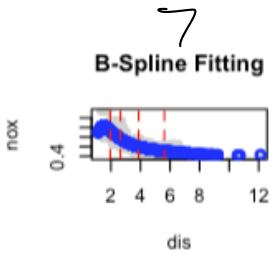
5



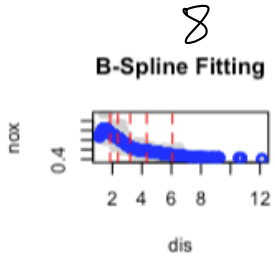
6



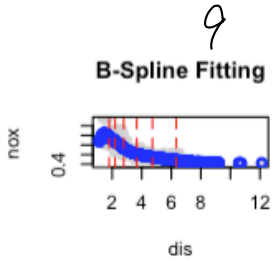
7



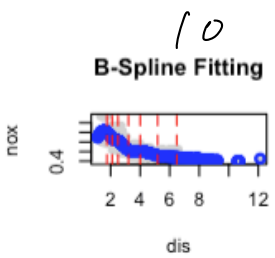
8



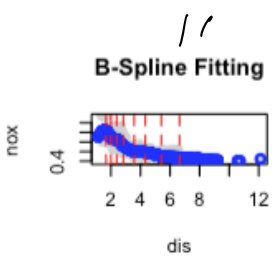
9



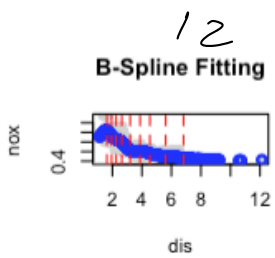
10



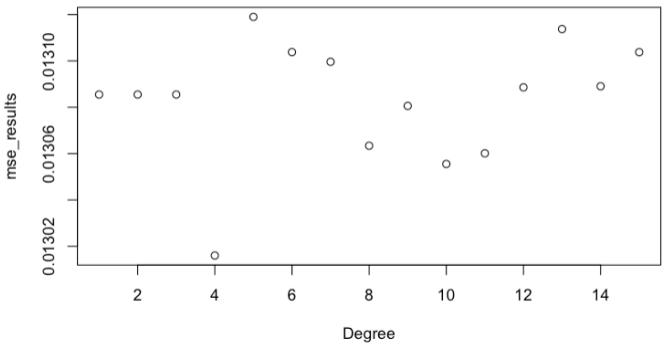
11



12



f)



```
library(ISLR2)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
library(splines)
```

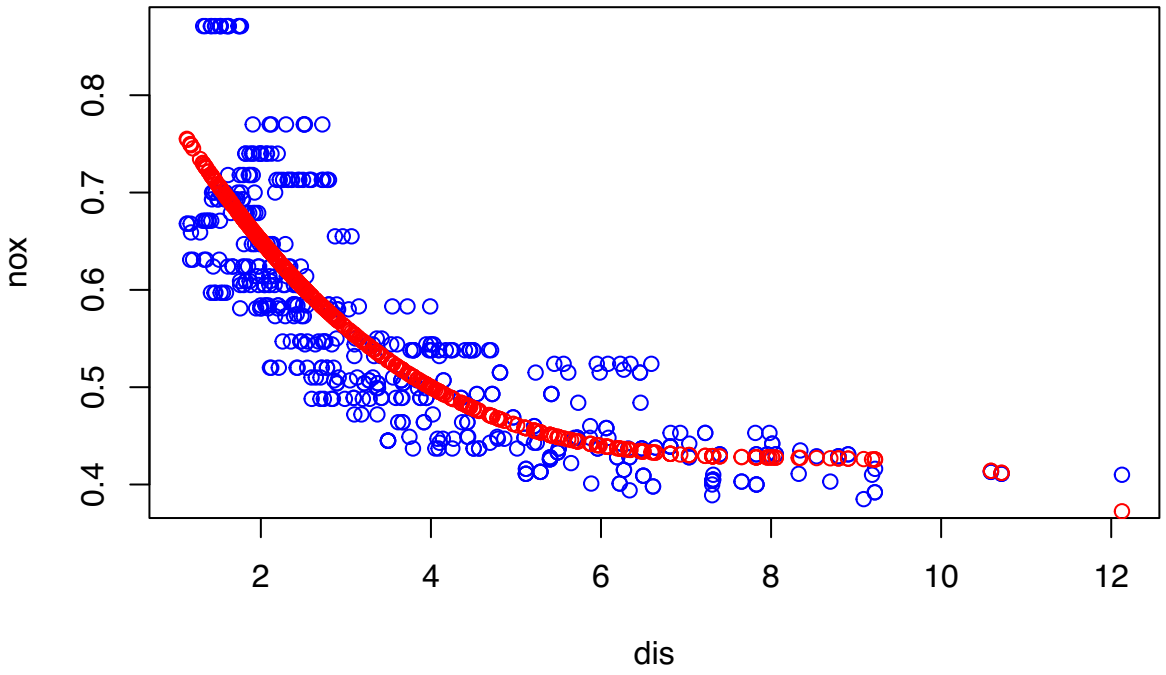
```
data = Boston
dis = data$dis
nox = data$nox
set.seed(666)
```

#a

```
model1 = lm(nox ~ poly(dis ,degree = 3))
model1
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, degree = 3))
##
## Coefficients:
##            (Intercept)  poly(dis, degree = 3)1  poly(dis, degree = 3)2
##                 0.5547                 -2.0031                  0.8563
## poly(dis, degree = 3)3
##                -0.3180
```

```
plot(dis,nox,col = "blue")
points(dis, fitted.values(model1), col = "red")
```



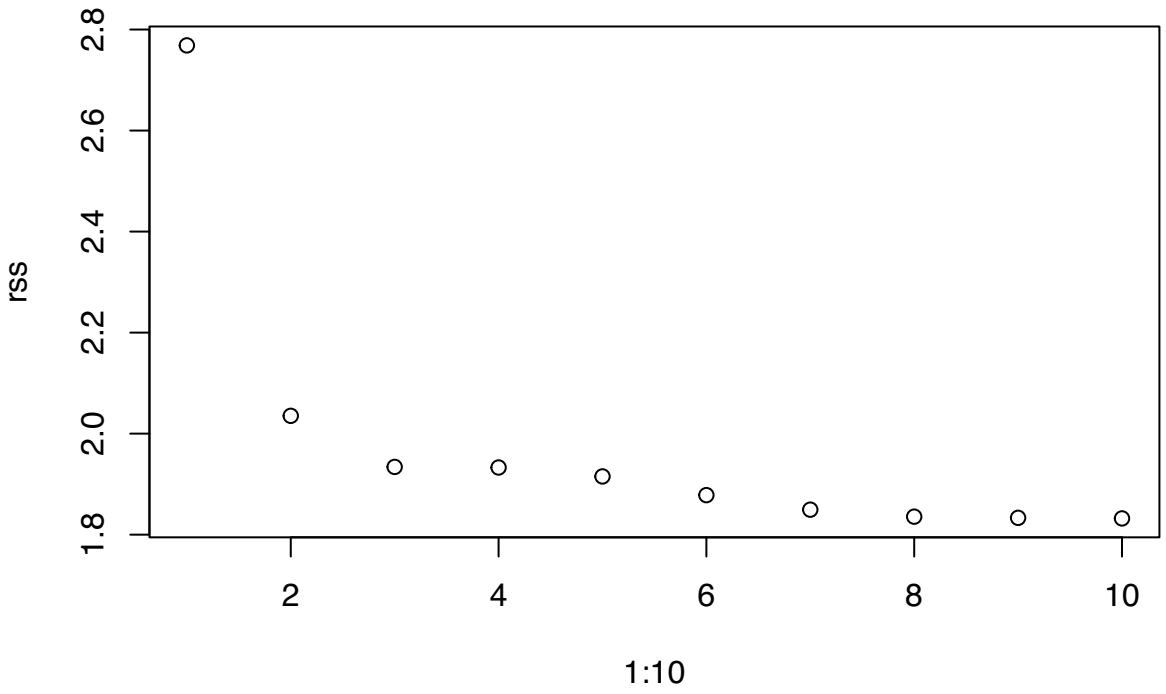
#b

```
rss = c()
for (i in (1:10)){
```

```
  modelx = lm(nox ~ poly(dis, degree = i))
  rss[i] = sum(residuals(modelx)^2)
}
rss
```

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

```
plot(1:10,rss)
```

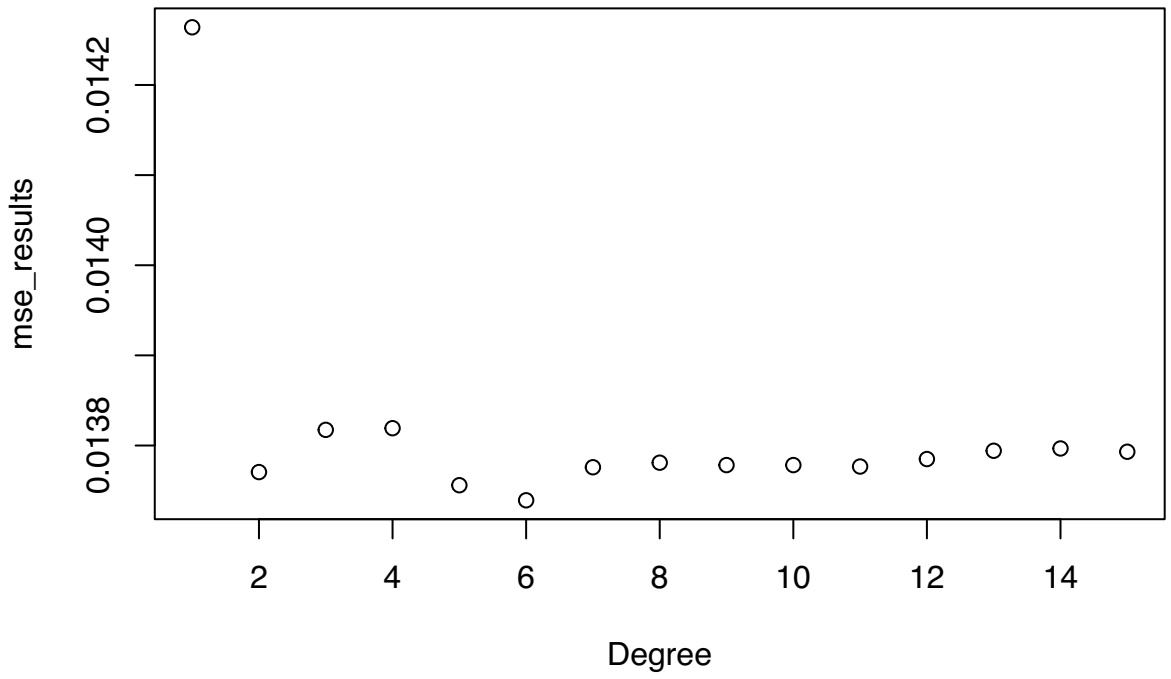


#c

```
folds = createFolds(nox, k = 5)
mse_results = c()
for (i in 1:15) {
  fold_mse = c()
  for (j in 1:5) {
    train_indices = unlist(folds[-j])
    test_indices = unlist(folds[j])
    modelx = lm(nox[train_indices] ~ poly(dis[train_indices], degree = i))
    predictions = predict(modelx, data.frame(dis[test_indices]))
    fold_mse[j] = mean((predictions - nox[test_indices])^2)
  }
  mse_results[i] = mean(fold_mse)
}
mse_results
```

```
##  [1] 0.01426399 0.01377050 0.01381735 0.01381920 0.01375593 0.01373921
##  [7] 0.01377586 0.01378093 0.01377826 0.01377827 0.01377667 0.01378497
## [13] 0.01379417 0.01379671 0.01379306
```

```
plot((1:15), mse_results,xlab='Degree')
```



#d

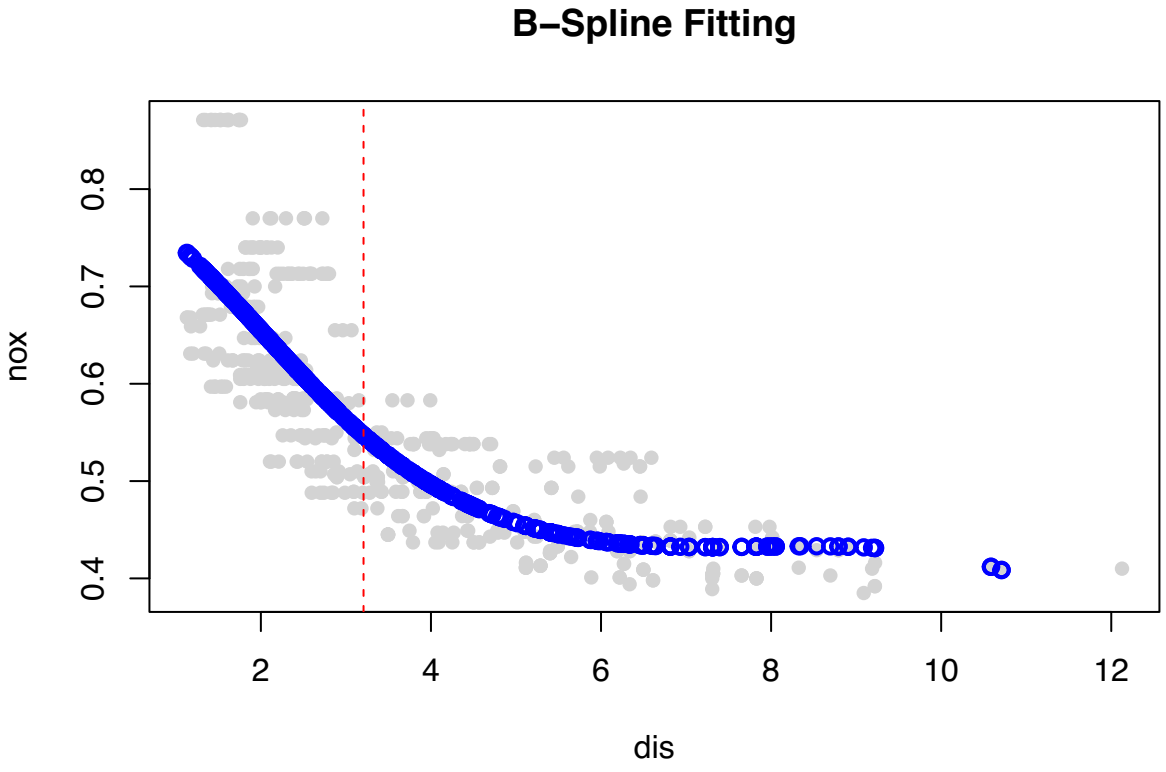
```
bs_s = bs(dis,df = 4)
model2 = lm(nox ~ bs_s)
model2
```

```
##
## Call:
## lm(formula = nox ~ bs_s)
##
## Coefficients:
## (Intercept)        bs_s1        bs_s2        bs_s3        bs_s4
##      0.7345      -0.0581      -0.4636      -0.1998      -0.3888
```

```
knots = attr(bs_s, "knots")
knots
```

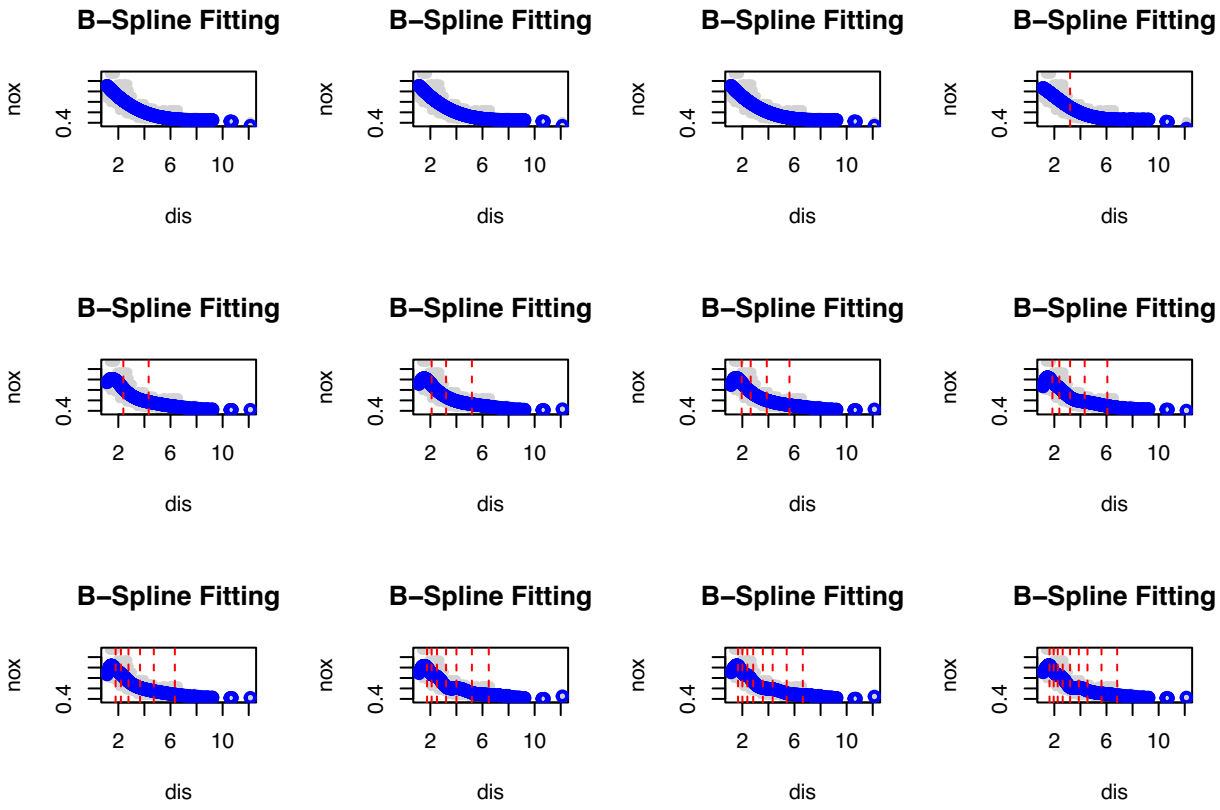
```
## [1] 3.20745
```

```
predictions = predict(model2)
plot(dis, nox, main = "B-Spline Fitting", pch = 16, col = "lightgray")
points(dis, predictions, col = "blue", lwd = 2)
abline(v = knots, col = 'red',lty = 2)
```



#e

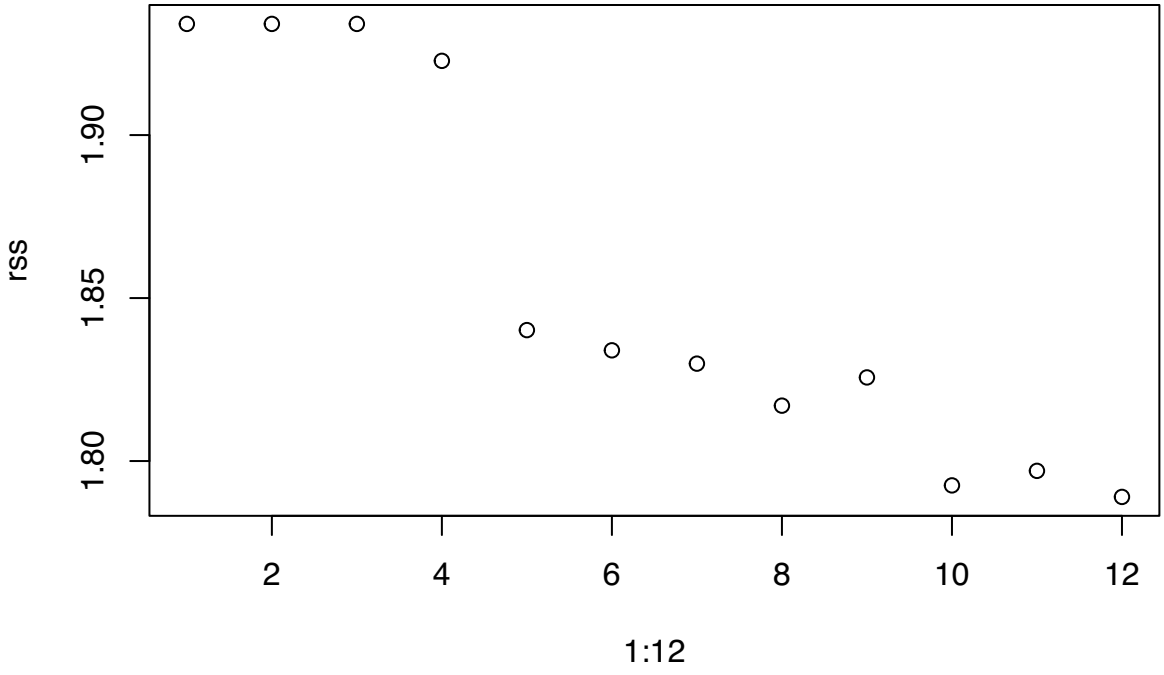
```
rss = c()
par(mfrow = c(3, 4))
for (i in (1:12)){
  bs_s = bs(dis,df = i)
  modelx = lm(nox ~ bs_s)
  rss[i] = sum(residuals(modelx)^2)
  predictions = predict(modelx)
  knots = attr(bs_s, "knots")
  plot(dis, nox, main = "B-Spline Fitting", pch = 16, col = "lightgray")
  points(dis, predictions, col = "blue", lwd = 2)
  abline(v = knots, col = 'red',lty = 2)
}
```




```
rss
```

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

```
par(mfrow = c(1, 1))
plot(1:12, rss)
```



#f

```
folds = createFolds(nox, k = 5)
mse_results = c()
for (i in 1:15) {
  fold_mse = c()
  for (j in 1:5) {
    train_indices = unlist(folds[-j])
    test_indices = unlist(folds[j])
    bs_s = bs(dis[train_indices],df = i)
    modelx = lm(nox[train_indices] ~ bs_s)
    predictions = predict(modelx, data.frame(dis[test_indices]))
    fold_mse[j] = mean((predictions - nox[test_indices])^2)
  }
  mse_results[i] = mean(fold_mse)
}
mse_results
```

```
##  [1] 0.01308549 0.01308549 0.01308549 0.01301602 0.01311899 0.01310380
##  [7] 0.01309962 0.01306340 0.01308059 0.01305554 0.01306012 0.01308859
## [13] 0.01311373 0.01308906 0.01310378
```

```
plot((1:15), mse_results,xlab='Degree')
```