STATS 415 Homework 7

Yuzhou Peng

2024-03-29

```
library(ISLR2)
library(boot)
library(splines)
```

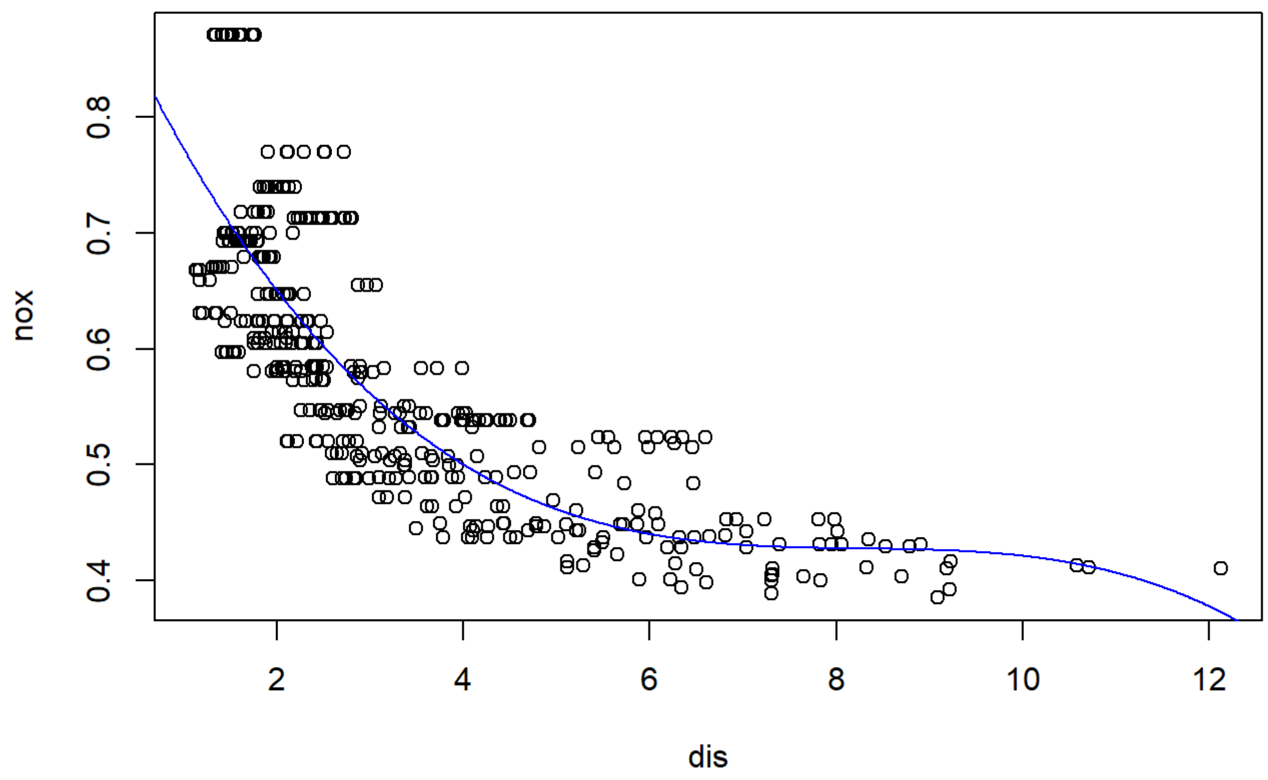
Problem 1

(a)

```
model_a <- lm(nox ~ poly(dis, 3, raw = T), data = Boston)
summary(model_a)
```

```
##
## Call:
## lm(formula = nox ~ poly(dis, 3, raw = T), 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)             0.9341281  0.0207076  45.110  < 2e-16 ***
## poly(dis, 3, raw = T)1 -0.1820817  0.0146973 -12.389  < 2e-16 ***
## poly(dis, 3, raw = T)2  0.0219277  0.0029329   7.476 3.43e-13 ***
## poly(dis, 3, raw = T)3 -0.0008850  0.0001727  -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
model_formula <- function(x, degree){
  mod <- lm(nox ~ poly(dis, degree, raw = T), data = Boston)
  X <- c()
  model_coef <- as.numeric(coef(mod))
  for (i in 1:length(model_coef)){
    predictor <- x^(i-1)
    X[i] = predictor
  }
  func_val <- sum(model_coef * X)
  return(func_val)
}

step <- seq(0,15, by = 0.01)



plot(Boston$dis, Boston$nox, xlab = "dis", ylab = "nox")

model_a_step <- c()
for (i in 1:length(step)){
  model_a_val <- model_formula(step[i], 3)
  model_a_step[i] <- model_a_val
}

lines(step, model_a_step, col = "blue")
```

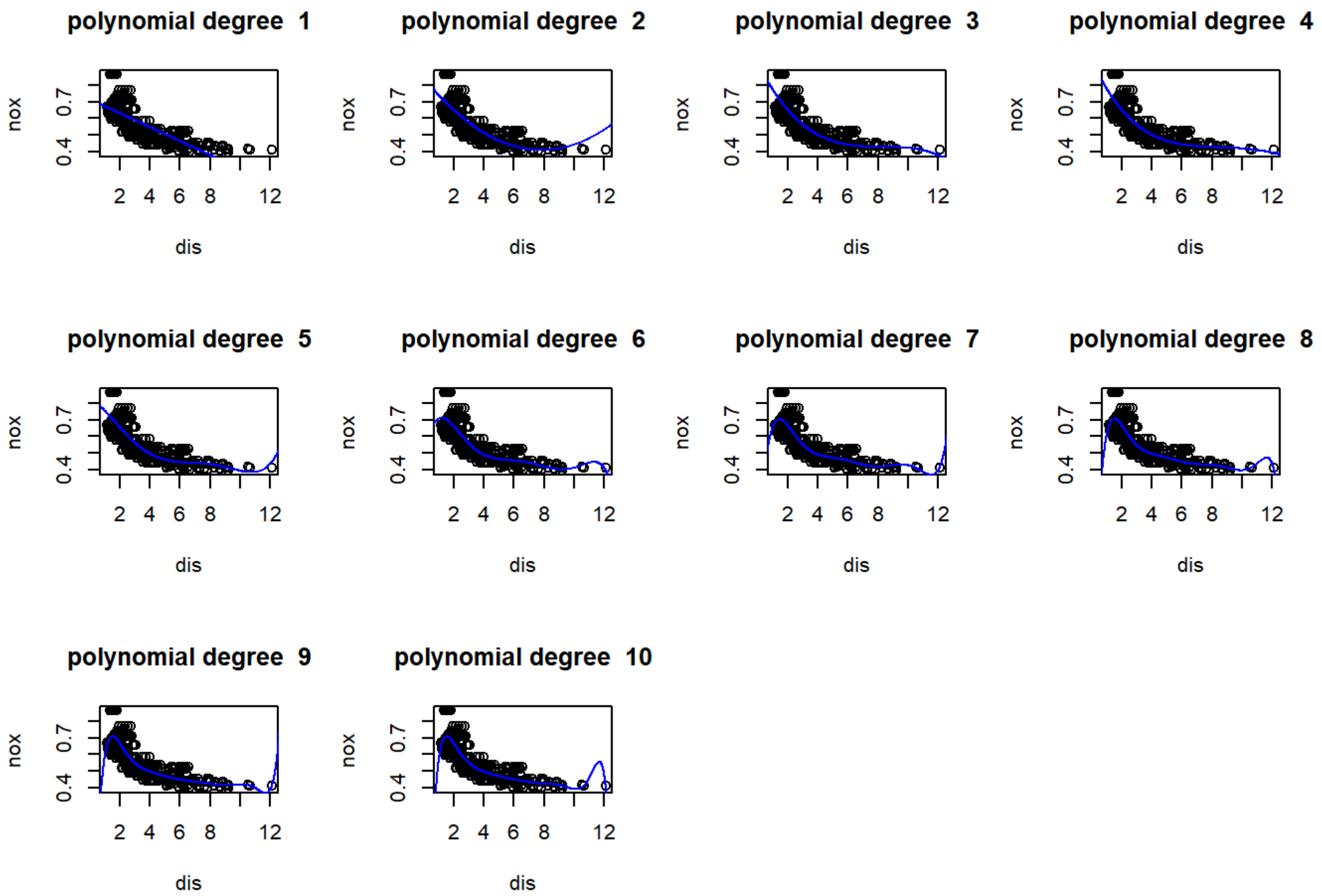


The output from summary suggest every coefficient is significant.
Multiple R-sqaured is 0.7148 which means the model is a good fit.

(b)

```
par(mfrow = c(3,4))


for (j in 1:10){
  model_step <- c()
  for (i in 1:length(step)){
    model_val <- model_formula(step[i], j)
    model_step[i] <- model_val
  }
  plot(Boston$dis, Boston$nox, xlab = "dis", ylab = "nox", main = paste("polynomial degree ",
j))
  lines(step, model_step , col = "blue")
}

for (i in 1:10){
  mod <- lm(nox ~ poly(dis, i, raw = T), data = Boston)
  print(paste("The RSS of polynomial model with degree", i, "is", deviance(mod) ))
}
```

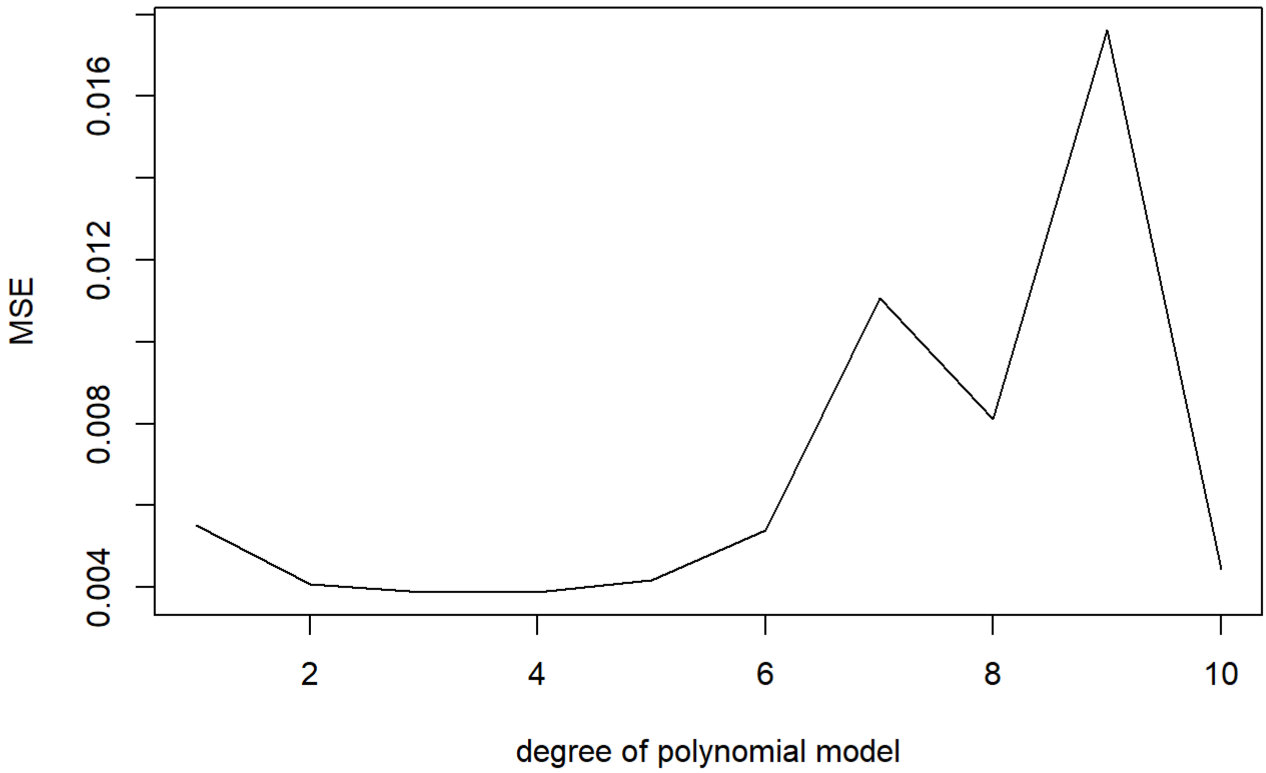
```
## [1] "The RSS of polynomial model with degree 1 is 2.76856285896928"
## [1] "The RSS of polynomial model with degree 2 is 2.03526186893526"
## [1] "The RSS of polynomial model with degree 3 is 1.93410670717907"
## [1] "The RSS of polynomial model with degree 4 is 1.93298132729859"
## [1] "The RSS of polynomial model with degree 5 is 1.91528996108431"
## [1] "The RSS of polynomial model with degree 6 is 1.87825729850818"
## [1] "The RSS of polynomial model with degree 7 is 1.84948361458295"
## [1] "The RSS of polynomial model with degree 8 is 1.83562968906756"
## [1] "The RSS of polynomial model with degree 9 is 1.8333308044916"
## [1] "The RSS of polynomial model with degree 10 is 1.83217112393134"
```



To avoid confusion, we plot 10 polynomial fits separately.
The RSS associated with model with degree increasing from 1 to 3 drops sharply from 2.76 to 1.93.
However, when degree >= 3, RSS drops very slowly, from 1,93 to 1.83.

(c)

```
study_data <- data.frame(dis = Boston$dis,
                         nox = Boston$nox)


cv_result <- c()
for (j in 1:10) {
  err_cv <- c()
  for (i in 1:nrow(study_data)) {
    train <- study_data[-i,]
    test <- study_data[i,]
    model_cv <- lm(nox ~ poly(dis, j, raw = T), data = train)
    test_fit <- as.numeric(predict(model_cv, test))
    err_sq_cv <- (test_fit - study_data$nox[i])^2
    err_cv[i] <- err_sq_cv
  }
  cv_result[j] <- mean(err_cv)
}


plot(cv_result, type = "l", xlab = "degree of polynomial model", ylab = "MSE")
```




```
which.min(cv_result)
```

```
## [1] 3
```

We apply LOOCV for cross validation. The result gives us 3 as the optimal degree of polynomial to fit.

(d)

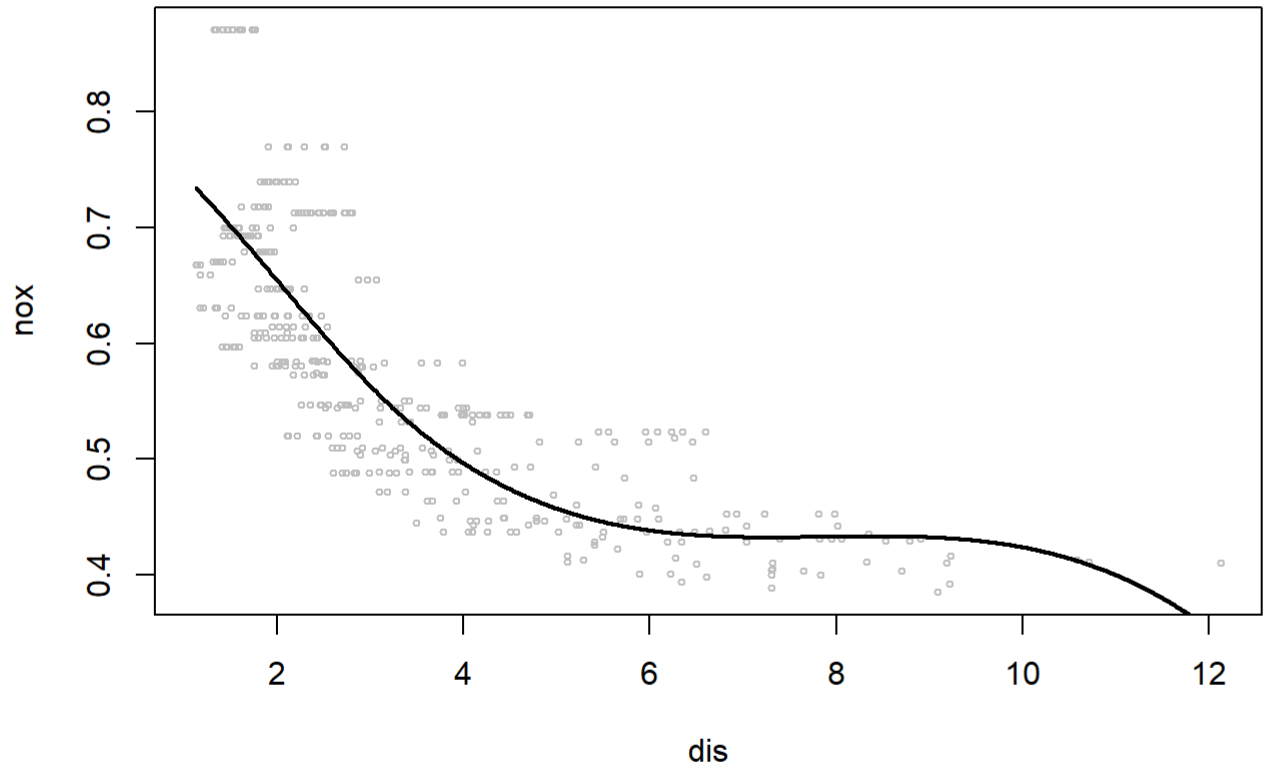
```
dis.grid <- seq(from = min(Boston$dis), to = max(Boston$dis), by = 0.01)

splines_d <- lm(nox ~ bs(dis, df = 4), data = study_data)
summary(splines_d)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis, df = 4), data = study_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124622 -0.039259 -0.008514  0.020850  0.193891
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.73447    0.01460  50.306  < 2e-16 ***
## bs(dis, df = 4)1 -0.05810    0.02186  -2.658  0.00812 **
## bs(dis, df = 4)2 -0.46356    0.02366 -19.596  < 2e-16 ***
## bs(dis, df = 4)3 -0.19979    0.04311  -4.634 4.58e-06 ***
## bs(dis, df = 4)4 -0.38881    0.04551  -8.544  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06195 on 501 degrees of freedom
## Multiple R-squared:  0.7164, Adjusted R-squared:  0.7142
## F-statistic: 316.5 on 4 and 501 DF,  p-value: < 2.2e-16
```

```
deviance(splines_d)
```

```
## [1] 1.922775
```

```
preds <- predict(splines_d, newdata = data.frame(dis = dis.grid))
plot(study_data$dis, study_data$nox, cex = .5, col = "grey",
xlab = "dis", ylab = "nox")
lines(dis.grid, preds, lwd = 2)
```

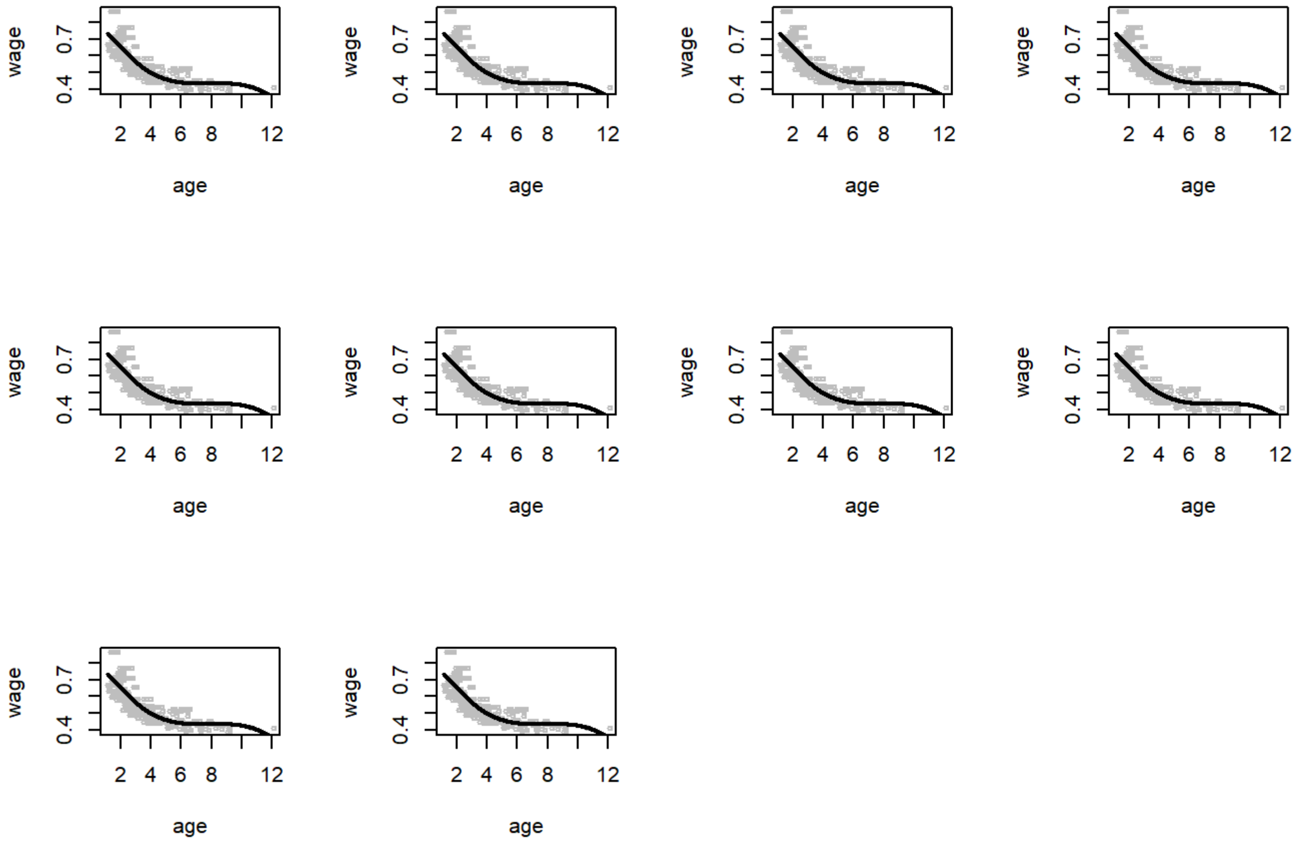


The output from summary suggest every coefficient is significant.
Multiple R-sqaured is 0.7164 which means the model is a good fit.
With df = 4, there are 3 knots. From the plot, we put the knots on roughly dis = 4, dis = 6 and dis = 8.

(e)

```
par(mfrow = c(3,4))

for (i in 3:12){
  splines <- lm(nox ~ bs(dis, df = i), data = study_data)
  preds <- predict(splines_d, newdata = data.frame(dis = dis.grid))
  plot(study_data$dis, study_data$nox, cex = .5, col = "grey",
  xlab = "age", ylab = "wage")
  lines(dis.grid, preds, lwd = 2)
}

for (i in 3:12){
  mod <- lm(nox ~ bs(dis, df = i), data = Boston)
  print(paste("The RSS splines model with degree of freedom", i, "is", deviance(mod) ))
}
```

```
## [1] "The RSS splines model with degree of freedom 3 is 1.93410670717907"
## [1] "The RSS splines model with degree of freedom 4 is 1.92277499281193"
## [1] "The RSS splines model with degree of freedom 5 is 1.84017280148852"
## [1] "The RSS splines model with degree of freedom 6 is 1.83396590316021"
## [1] "The RSS splines model with degree of freedom 7 is 1.82988444592328"
## [1] "The RSS splines model with degree of freedom 8 is 1.81699505672523"
## [1] "The RSS splines model with degree of freedom 9 is 1.82565251038706"
## [1] "The RSS splines model with degree of freedom 10 is 1.79253488955613"
## [1] "The RSS splines model with degree of freedom 11 is 1.79699182173143"
## [1] "The RSS splines model with degree of freedom 12 is 1.78899914528888"
```



We select degree of freedom from 3 to 12 to plot because the bs() function output 3 as the smallest number of df.
The RSS drops slowly from 1.93 to 1.79 as df increasing from 3 to 12.

(f)

```
cv_result <- c()
for (j in 3:12) {
  err_cv <- c()
  for (i in 1:nrow(study_data)) {
    train <- study_data[-i,]
    test <- study_data[i,]
    model_cv <- lm(nox ~ bs(dis, df = j), data = train)
    test_fit <- as.numeric(predict(model_cv, test))
    err_sq_cv <- (test_fit - study_data$nox[i])^2
    err_cv[i] <- err_sq_cv
  }
  cv_result[j] <- mean(err_cv)
}
```

```
## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.1296,
## : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = numeric(0), Boundary.knots = c(1.137, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = 3.1992, Boundary.knots = c(1.1296, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = 3.2157, Boundary.knots = c(1.137, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(2.3817, 4.2673), Boundary.knots =
## c(1.1296, : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(2.3887, 4.3549), Boundary.knots =
## c(1.137, : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(2.1, 3.1992, 5.118), Boundary.knots =
## c(1.1296, : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(2.1007, 3.2157, 5.2119),
## Boundary.knots = c(1.137, : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.94984, 2.62334, 3.85838, 5.5714:
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.9512, 2.6439, 3.87584, 5.62168:
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.8498, 2.3817, 3.1992, 4.2673, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.8589, 2.3887, 3.2157, 4.3549, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.7912, 2.1974, 2.7778, 3.665, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.794, 2.198, 2.7778, 3.6659, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.7494, 2.1, 2.5052, 3.1992, 3.9986,
## : 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.7523, 2.1007, 2.5091, 3.2157, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

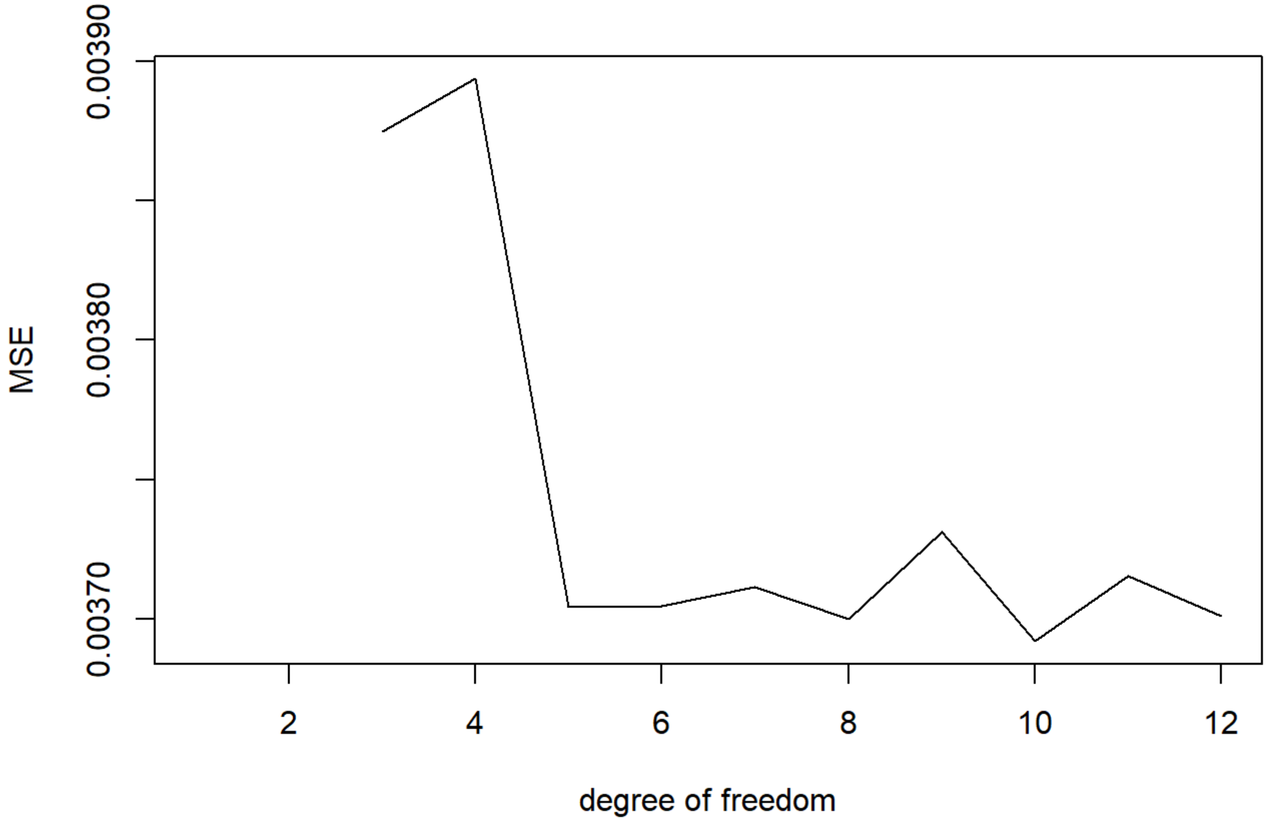
```
## Warning in bs(dis, degree = 3L, knots = c(1.6687, 2.0026, 2.3817, 2.829, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.6768, 2.0048, 2.3887, 2.834, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.62728, 1.94984, 2.25848, 2.62334, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
## Warning in bs(dis, degree = 3L, knots = c(1.63564, 1.9512, 2.26178, 2.6439, :
## 一些在结值界外的'x'数据有可能会引起病态底数
```

```
plot(cv_result, type = "l", xlab = "degree of freedom", ylab = "MSE")
```




```
which.min(cv_result)
```

```
## [1] 10
```

We apply LOOCV for cross validation.

The CV process gives 10 as the optimal df to fit the splines model.