Quiz4

Nam Jun Lee

12/02/2021

This question uses the variables dis(the weighted mean of distances to five boston employment centers) and nox(nitrogen oxide concentrations in parts per 10 million) from the Boston data in the ISLR package. We will treat dis as the predictor and nox as response.

```
# Boston dataset into bt variable
bt <- MASS::Boston
# summary of bt dataset
summary(bt)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm              age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

Q1. Use the poly() function to fit polynomial regression to the above data. Use a range of different polynomial degrees and choose the best fitting degree by using the anova() function. (Recall that all models being used here are nested). Compute $MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2$ for the best fitting model.

```
# fit the range of different polynomial degrees
fit_poly1 <- lm(nox ~ poly(dis, 3, raw = T), data = bt)
fit_poly2 <- lm(nox ~ poly(dis, 4, raw = T), data = bt)
fit_poly3 <- lm(nox ~ poly(dis, 5, raw = T), data = bt)
fit_poly4 <- lm(nox ~ poly(dis, 6, raw = T), data = bt)
fit_poly5 <- lm(nox ~ poly(dis, 7, raw = T), data = bt)
fit_poly6 <- lm(nox ~ poly(dis, 8, raw = T), data = bt)
fit_poly7 <- lm(nox ~ poly(dis, 9, raw = T), data = bt)
fit_poly8 <- lm(nox ~ poly(dis, 10, raw = T), data = bt)
# find best fitting degree
anova(fit_poly1, fit_poly2, fit_poly3, fit_poly4, fit_poly5, fit_poly6, fit_poly7,
    fit_poly8, test = "F")
```

```
## Analysis of Variance Table
##
## Model 1: nox ~ poly(dis, 3, raw = T)
## Model 2: nox ~ poly(dis, 4, raw = T)
## Model 3: nox ~ poly(dis, 5, raw = T)
## Model 4: nox ~ poly(dis, 6, raw = T)
## Model 5: nox ~ poly(dis, 7, raw = T)
## Model 6: nox ~ poly(dis, 8, raw = T)
## Model 7: nox ~ poly(dis, 9, raw = T)
## Model 8: nox ~ poly(dis, 10, raw = T)
##   Res.Df    RSS Df Sum of Sq       F   Pr(>F)
## 1    502 1.9341
## 2    501 1.9330  1  0.001125  0.3040 0.581606
## 3    500 1.9153  1  0.017691  4.7797 0.029265 *
## 4    499 1.8783  1  0.037033 10.0052 0.001657 **
## 5    498 1.8495  1  0.028774  7.7738 0.005505 **
## 6    497 1.8356  1  0.013854  3.7429 0.053601 .
## 7    496 1.8333  1  0.002299  0.6211 0.431019
## 8    495 1.8322  1  0.001160  0.3133 0.575908
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

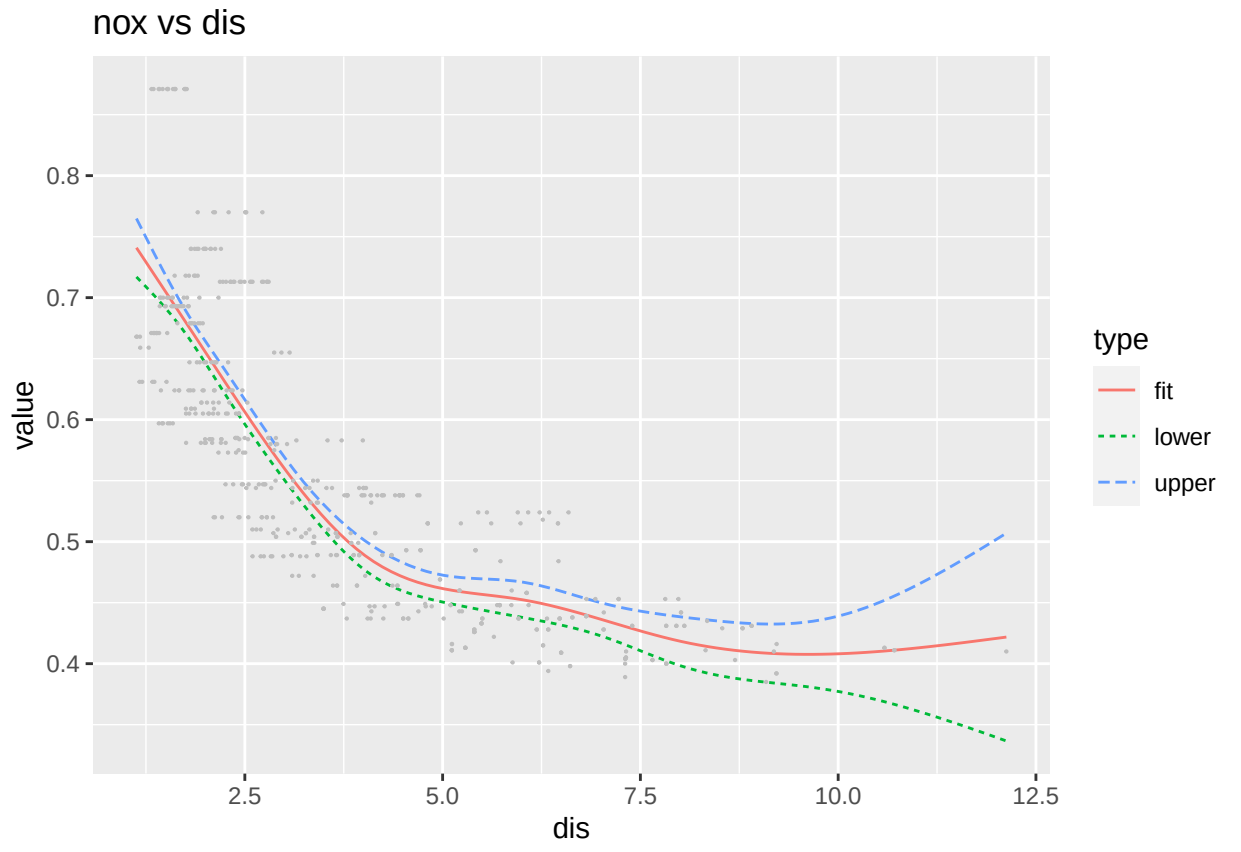
```
# compute MSE
mean(fit_poly4$residuals^2)
```

```
## [1] 0.003711971
```

As a result of evaluating the model using the range of eight polynomial orders, the p-value of Model 4 is most suitable at **0.001657**.
Model 4's MSE: **0.003711971**.

Q2. Fit a natural cubic spline to the above data using ns() function. Choose 4 equally spaced knots. Compute $MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2$ for this.

```
# ns() function to implement natural splines and 4 equally spaced knots
fit_spline <- lm(nox ~ ns(dis, knots = c(2, 4, 6, 8)), data = bt)
# degree of the spline will default to three
dis.lims <- range(bt$dis)
dis.grid <- seq(dis.lims[1], dis.lims[2], length.out = 100)
pred <- predict(fit_spline, newdata = list(dis = dis.grid), se = T)
# fit default
fitted <- pred$fit
lower <- fitted - 2 * pred$se.fit
upper <- fitted + 2 * pred$se.fit
numpred <- length(dis.grid)
df.spline <- data.frame(value = c(fitted, lower, upper), type = c(rep("fit", numpred),
    rep("lower", numpred), rep("upper", numpred)), dis = rep(dis.grid, 3))
# show graph
ggplot() + geom_line(data = df.spline, aes(x = dis, y = value, color = type, linetype = type)) +
    geom_point(data = bt, aes(x = dis, y = nox), size = 0.1, colour = "grey") + ggtitle("nox vs dis")
```



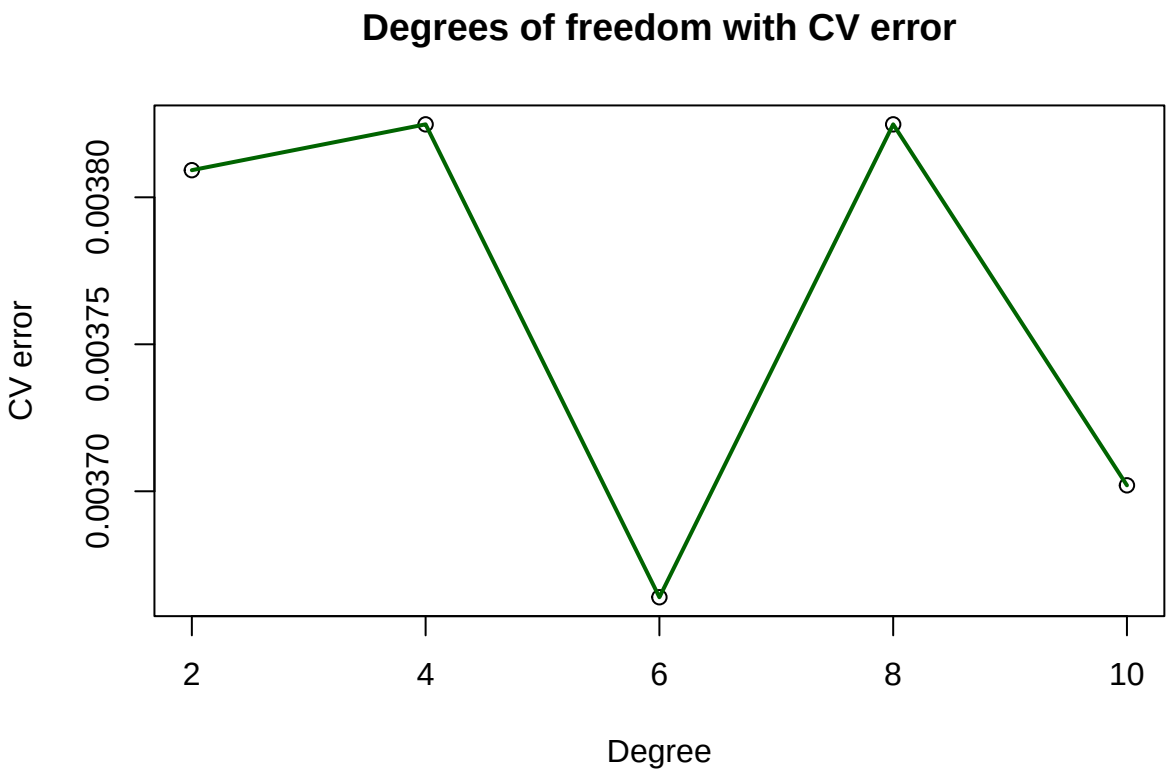

```
# compute MSE
mean(fit_spline$residuals^2)
```

```
## [1] 0.003761604
```

As a result of the fit by knots of the natural cubic spline to 2,4,6,8 it can be seen that the MSE is **0.003761604**.

Q3. Recall the idea of cross validation from earlier in the semester where the data is repeatedly broken in testing and training in order to compute a cross validation error. Perform a 5-fold cross validation over the number of knots (say, 2,4,6,8,10), to choose the best fitting natural cubic spline. Report MSE for the best fitting model.

```
# set seed
set.seed(1)
# set number of knots
knot <- c(2, 4, 6, 8, 10)
ran <- knot
# set mse range
cv.error <- rep(0, 5)
for (i in ran) {
    fit.cv.spline <- glm(nox ~ ns(dis, df = i), data = bt)
    cv.error[i - 5] = cv.glm(bt, fit.cv.spline, K = 5)$delta[1]  # compute cv error
}
# show plot
plot(ran, cv.error, xlab = "Degree", ylab = "CV error")
lines(ran, cv.error, lwd = 2, col = "darkgreen")
title(main = "Degrees of freedom with CV error")
```




```
# best fitting natural cubic spline
fit.cv.best <- lm(nox ~ ns(dis, knots = 6), data = bt)
# MSE
mean(fit.cv.best$residuals^2)
```

```
## [1] 0.004103866
```

As a result of performing 5-fold cross-validation after grouping knots to 2,4,6,8,10, it is most suitable because the cv error of **knot 6** was the lowest.
MSE: **0.004103866**.