HW9

Minh Luc, Devin Pham, Kyle Moore

Friday of Week 9, 05/27/2022

Contents

We all contributed equally for this homework.

Question 0

Member 1:

- Name: Minh Luc
- Student ID: A17209607

Member 2:

- Name: Kyle Moore
- Student ID: A14271413

Member 3:

- Name: Devin Pham
- Student ID: A17198936

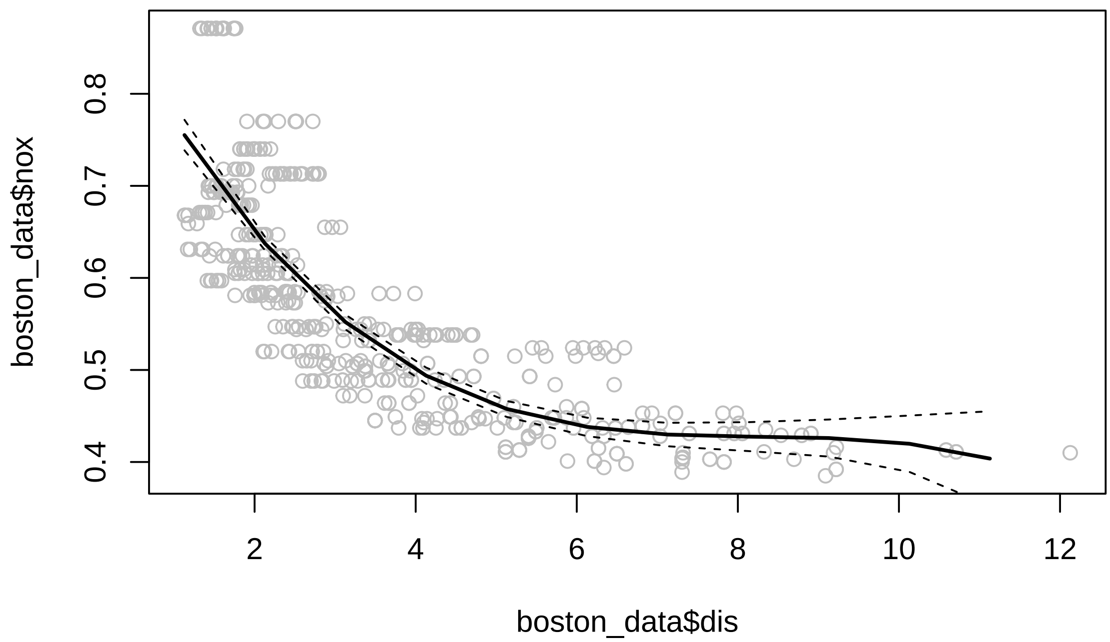
Question 1

- **(a)**

```
library(ISLR2)
library(splines)
boston_data <- Boston

fit <- lm(nox ~ bs(dis), data = boston_data)

dislims <- range(boston_data$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2])

pred <- predict(fit, newdata = list(dis = dis.grid), se = T)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ bs(dis), data = boston_data)
##
## 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)  0.755153   0.008283  91.168  < 2e-16 ***
## bs(dis)1    -0.498271   0.032542 -15.312  < 2e-16 ***
## bs(dis)2    -0.233520   0.036994  -6.312 6.05e-10 ***
## bs(dis)3    -0.382680   0.045455  -8.419 4.00e-16 ***
## ---
## 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
```

```
plot(boston_data$dis, boston_data$nox, col = "gray")
lines(dis.grid, pred$fit, lwd = 2)
lines(dis.grid, pred$fit + 2 * pred$se, lty = "dashed")
lines(dis.grid, pred$fit - 2 * pred$se, lty = "dashed")
```

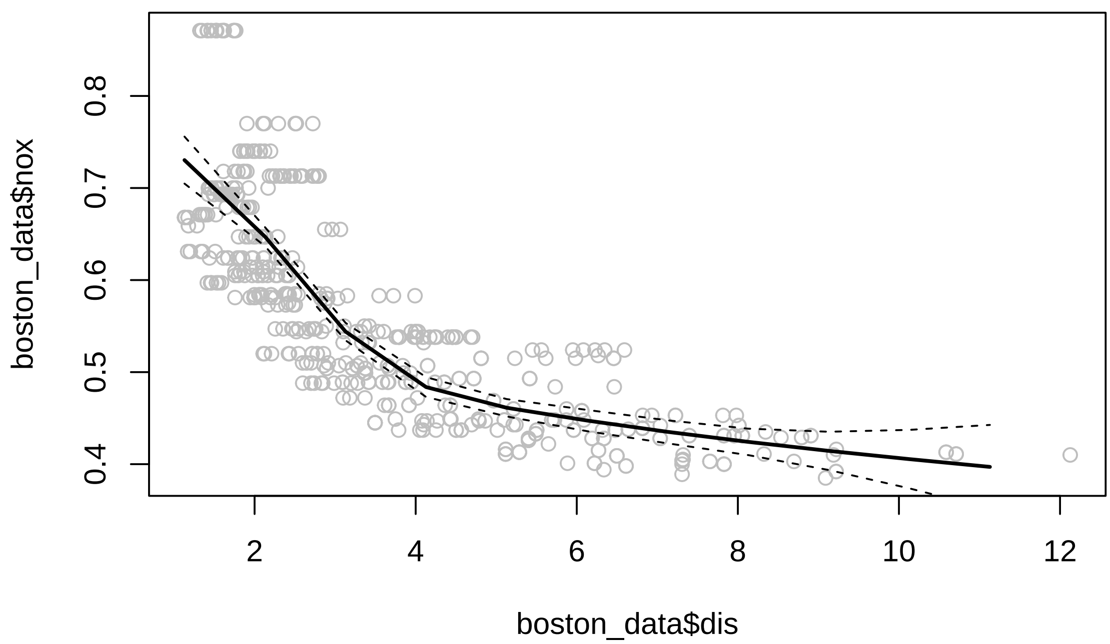


- **(b)**

```
fit <- lm(nox ~ ns(dis, df = 4), data = boston_data)

pred <- predict(fit, newdata = list(dis = dis.grid), se = T)
summary(fit)
```

```
##
## Call:
## lm(formula = nox ~ ns(dis, df = 4), data = boston_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12940 -0.04073 -0.00805  0.02494  0.19059
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.73032    0.01276   57.23   <2e-16 ***
## ns(dis, df = 4)1 -0.24312    0.01373  -17.70   <2e-16 ***
## ns(dis, df = 4)2 -0.27001    0.01724  -15.67   <2e-16 ***
## ns(dis, df = 4)3 -0.38799    0.03179  -12.21   <2e-16 ***
## ns(dis, df = 4)4 -0.30464    0.03105   -9.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06135 on 501 degrees of freedom
## Multiple R-squared:  0.7219, Adjusted R-squared:  0.7197
## F-statistic: 325.1 on 4 and 501 DF,  p-value: < 2.2e-16
```

```r
plot(boston_data$dis, boston_data$nox, col = "gray")
lines(dis.grid, pred$fit, lwd = 2)
lines(dis.grid, pred$fit + 2 * pred$se, lty = "dashed")
lines(dis.grid, pred$fit - 2 * pred$se, lty = "dashed")
```




```r
attr(terms(fit), "predvars") # show chosen knots
```

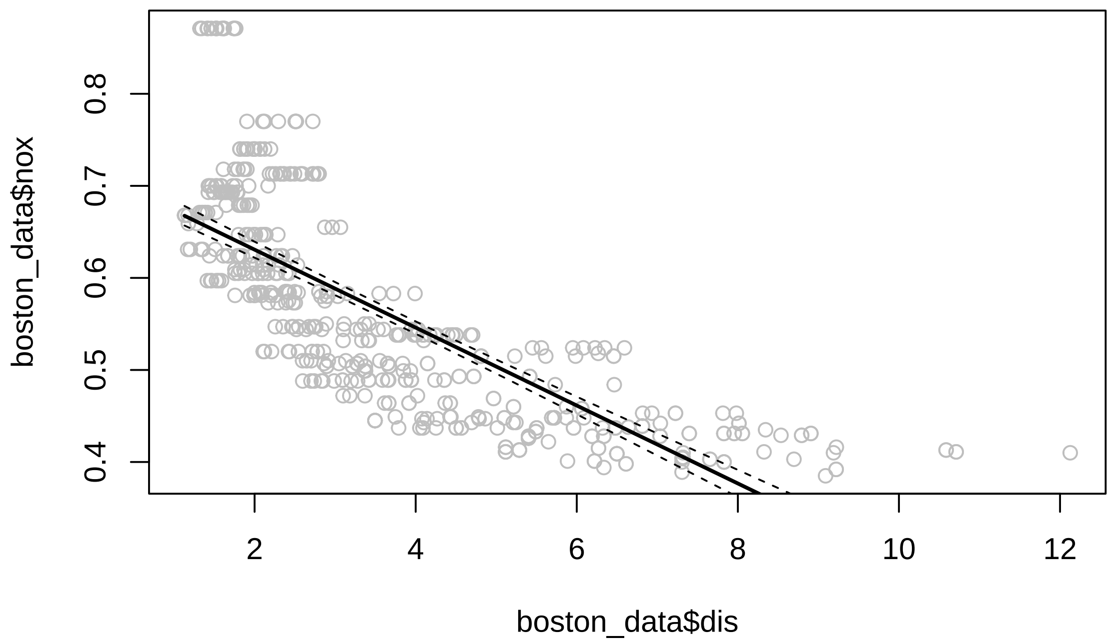
```
## list(nox, ns(dis, knots = c(`25%` = 2.100175, `50%` = 3.20745,
## `75%` = 5.188425), Boundary.knots = c(1.1296, 12.1265), intercept = FALSE))
```

Knots were chosen by ns() as the quartiles at 25, 50, and 75% of the data as seen above, and the boundary knots defaulted to the range of the data.

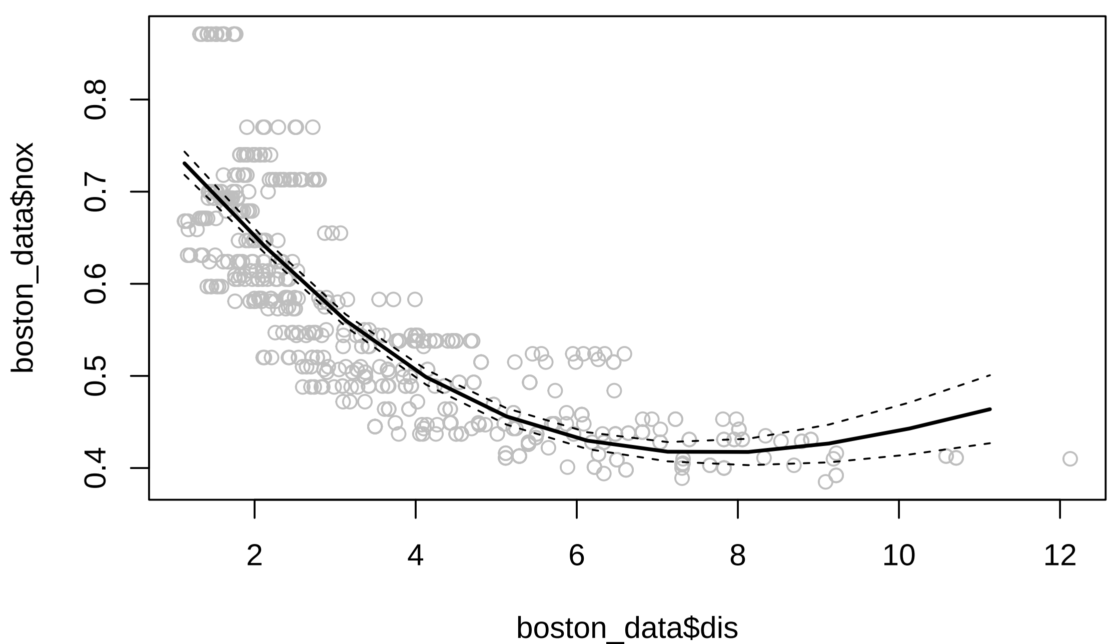
- **(c)**

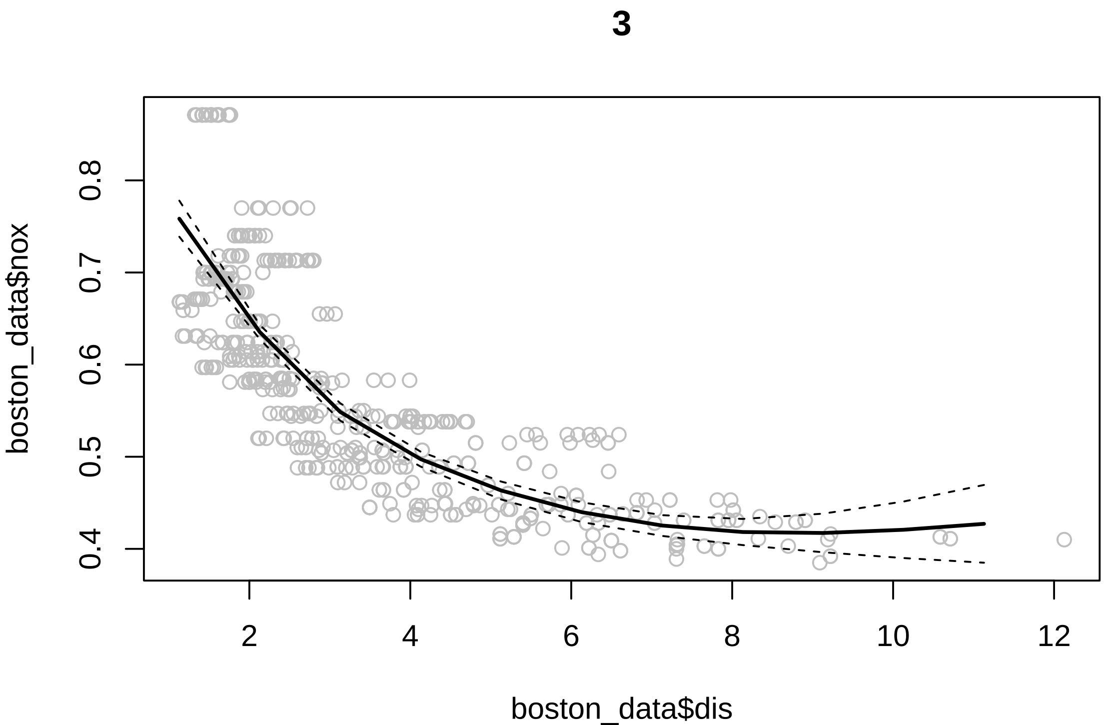
```r
rss <- list()
for (i in 1:10) {
  fit <- lm(nox ~ ns(dis, df = i), data = boston_data)
  pred <- predict(fit, newdata = list(dis = dis.grid), se = T)
  rss[i] <- sum((fit$residuals) ^ 2) # RSS

  plot(boston_data$dis, boston_data$nox, col = "gray", main = i)
  lines(dis.grid, pred$fit, lwd = 2)
  lines(dis.grid, pred$fit + 2 * pred$se, lty = "dashed")
  lines(dis.grid, pred$fit - 2 * pred$se, lty = "dashed")
}
```

1

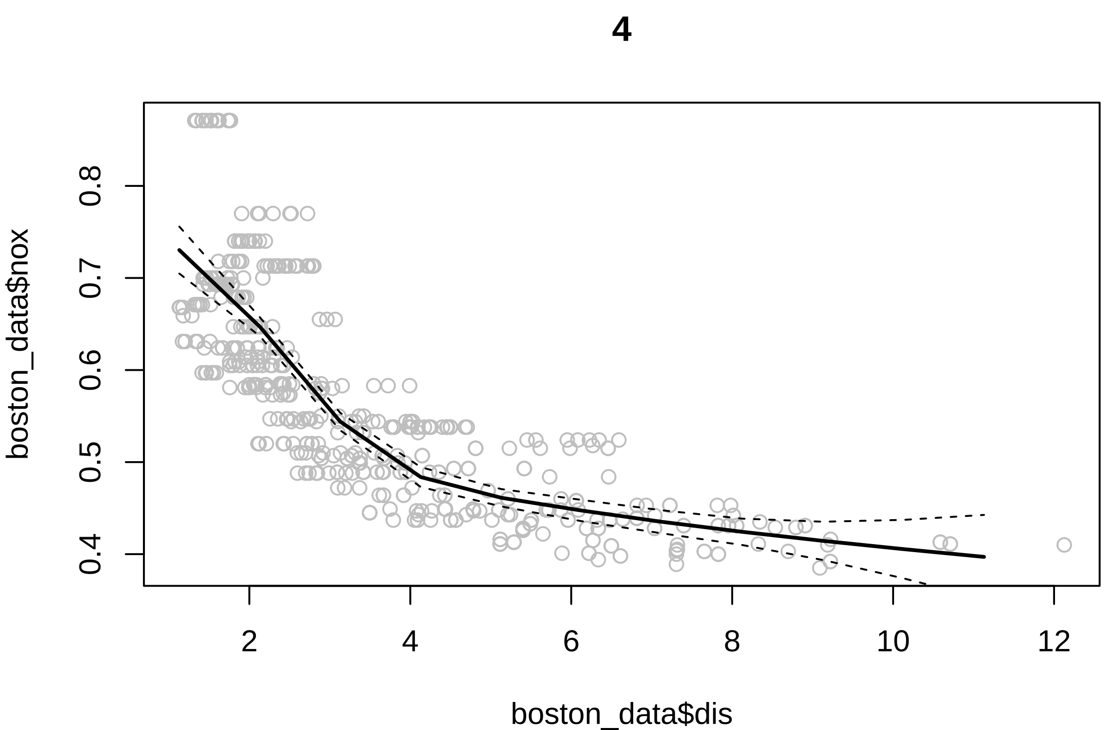

boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

2


boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

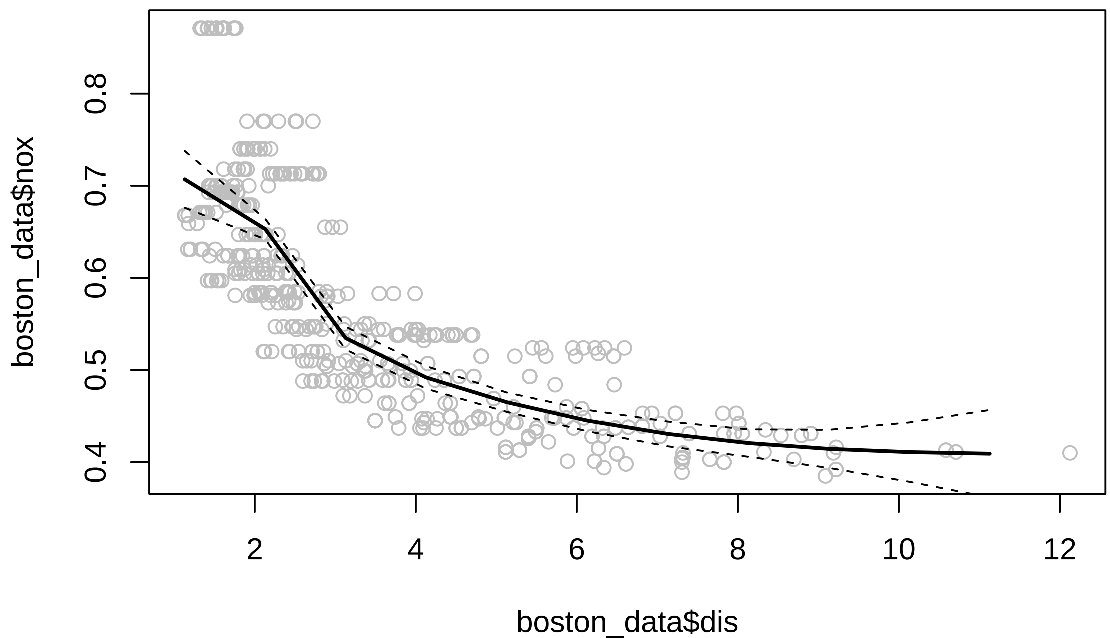


3
0.4
0.5
0.6
0.7
0.8
boston_data$nox
2
4
6
8
10
12
boston_data$dis



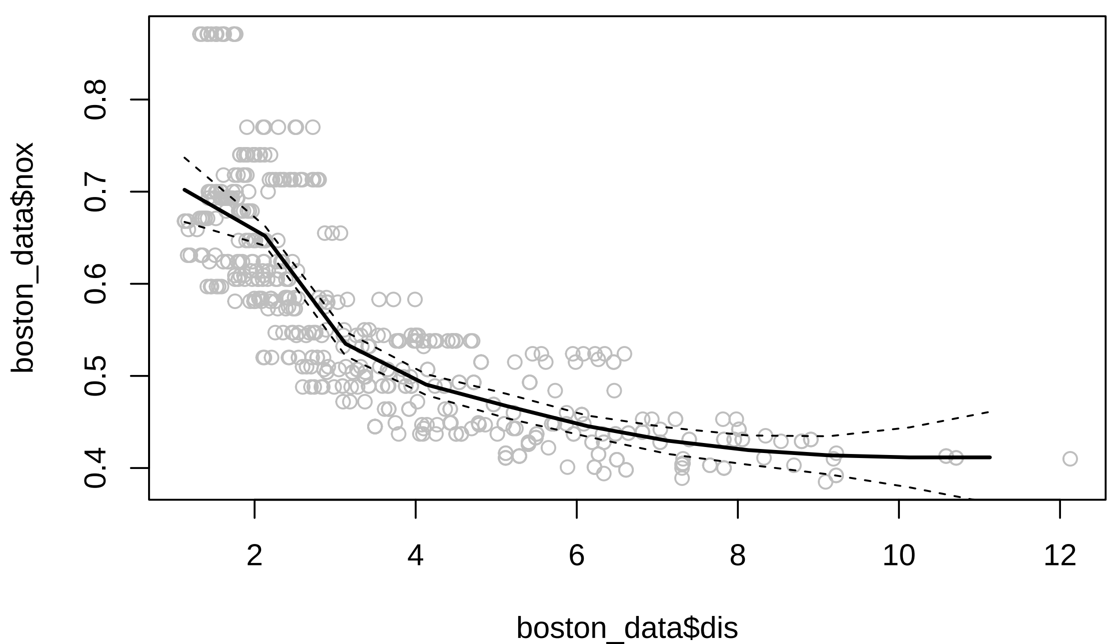
4
0.4
0.5
0.6
0.7
0.8
boston_data$nox
2
4
6
8
10
12
boston_data$dis

5



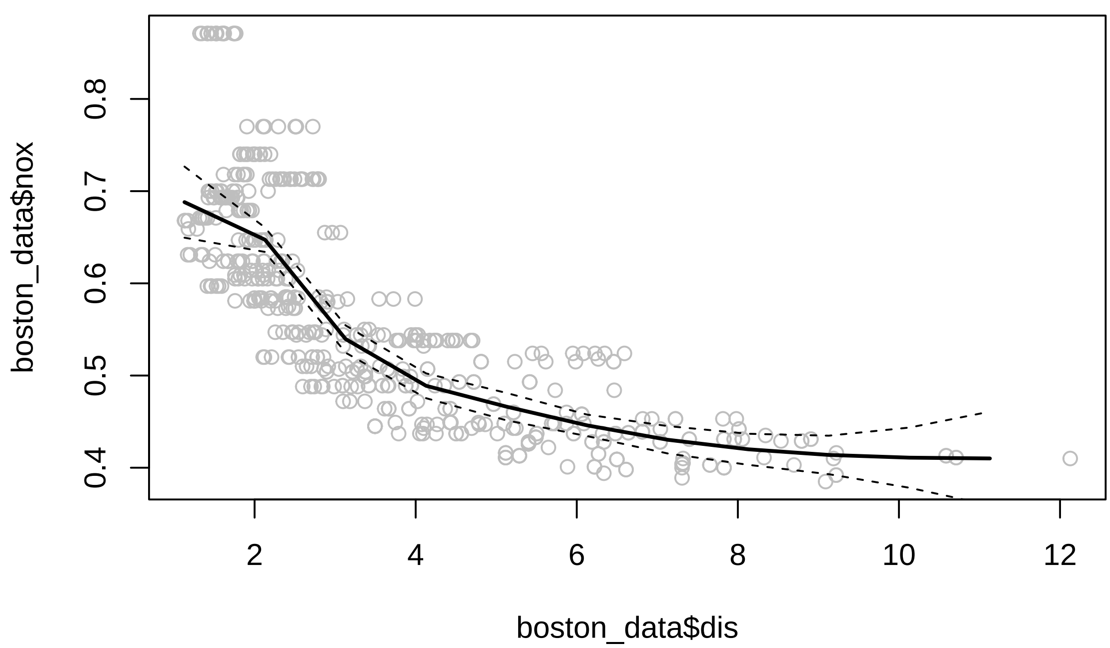
boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

6

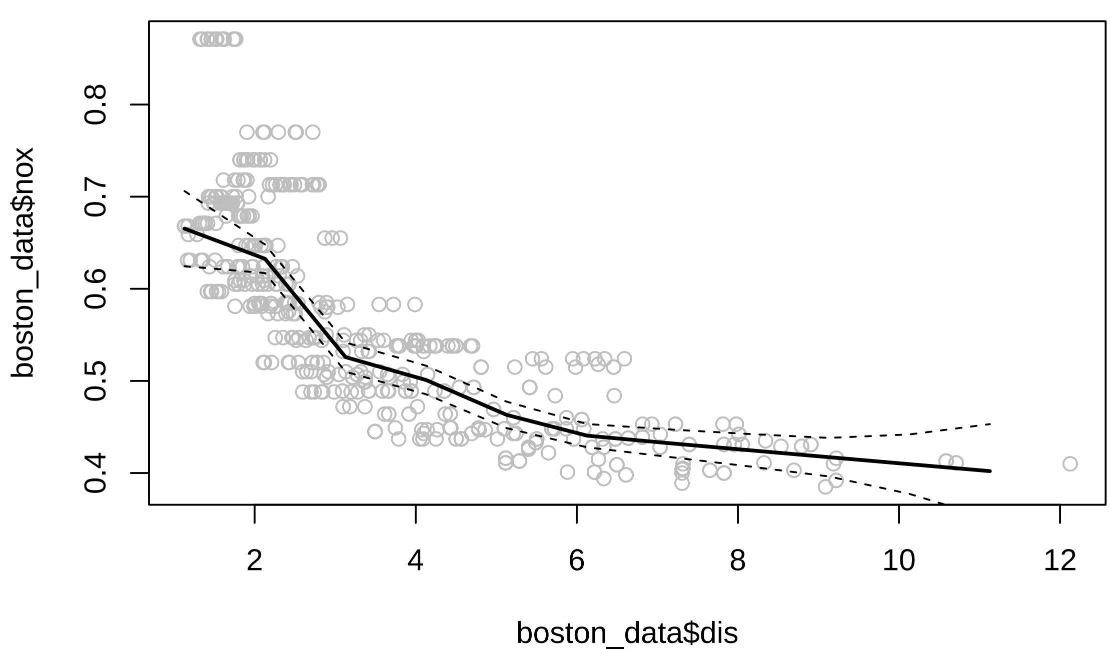


boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

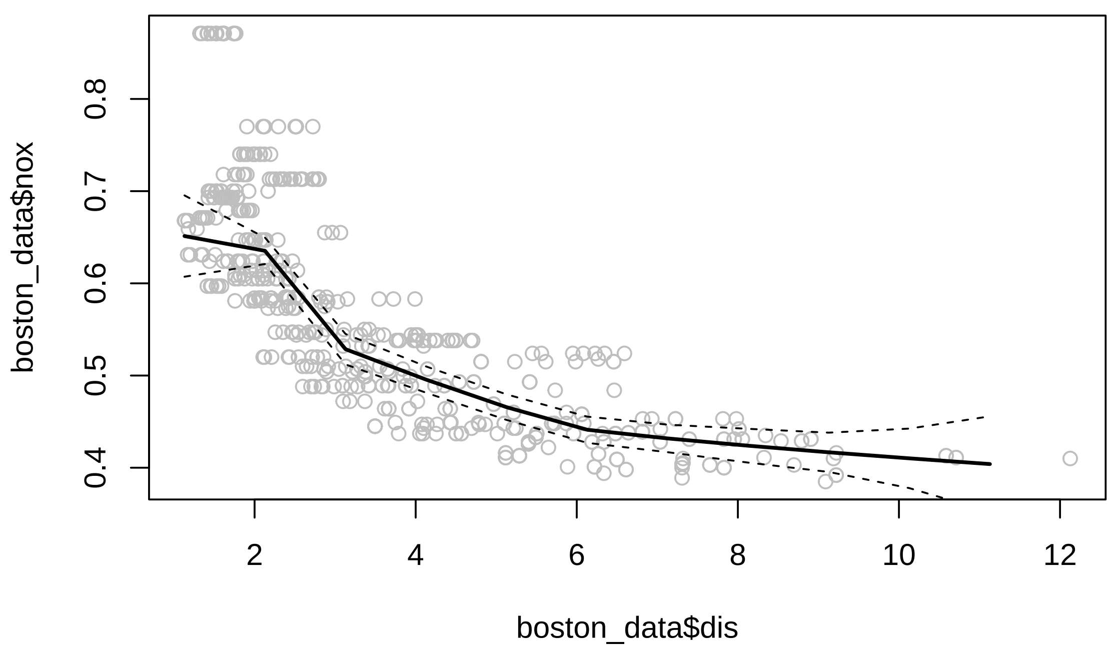
7


boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

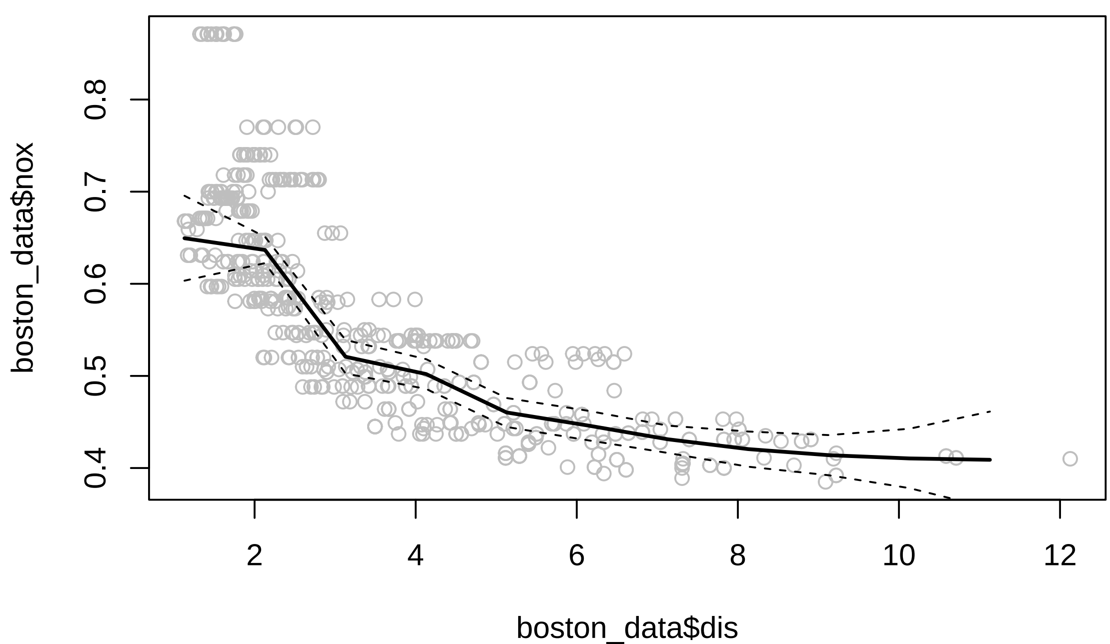
8


boston_data$nox
0.4
0.5
0.6
0.7
0.8
2
4
6
8
10
12
boston_data$dis

9



10



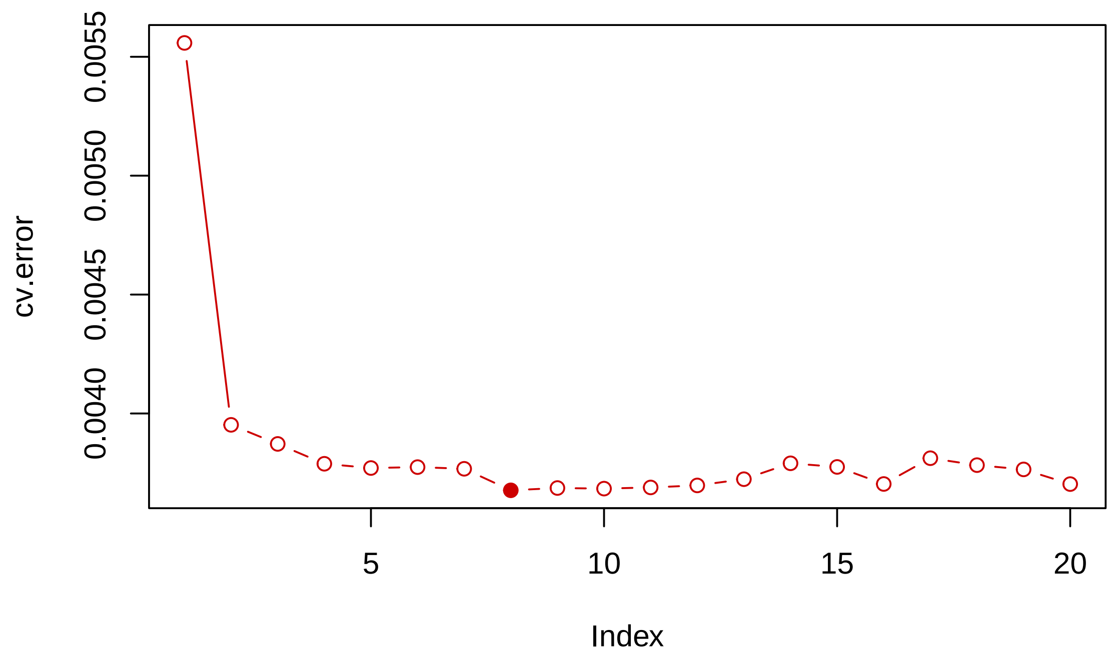

```
print(rss) # print the residual sum of squares for all dfs.
```

```
## [[1]]
## [1] 2.768563
##
## [[2]]
## [1] 1.974579
##
## [[3]]
## [1] 1.930501
##
## [[4]]
## [1] 1.885805
##
## [[5]]
## [1] 1.860232
##
## [[6]]
## [1] 1.854157
##
## [[7]]
## [1] 1.848602
##
## [[8]]
## [1] 1.797749
##
## [[9]]
## [1] 1.798482
##
## [[10]]
## [1] 1.789243
```

We can see that the higher the df/knots the better the fit to the data. This makes sense that an increase in model complexity increases accuracy on training data, but we must be careful for overfitting. 8-10 have similar RSS so I would be inclined to follow the parsimony principle and choose 8.

- **(d)**

```
library(boot)
set.seed(1)
cv.error = rep(0,20)
for(i in 1:20){
  glm.fit = glm(nox ~ ns(dis, df = i), data = boston_data)
  cv.error[i] = cv.glm(boston_data, glm.fit, K=10)$delta[1]
}
plot(cv.error,type="b",col="red3")
points(which.min(cv.error),min(cv.error),pch=16,col="red3")
```



Similar to our graphed results above, except here df = 8 is the lowest error and best choice. We can see that 8-12 are similar in error again as well.

- **(e)**

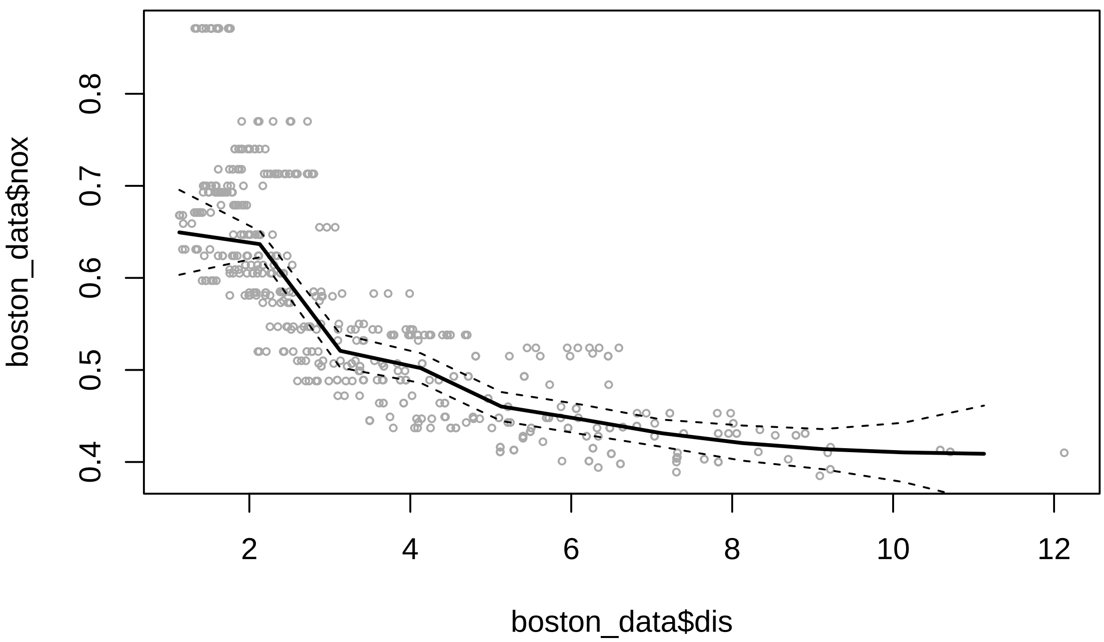
```
fit <- smooth.spline(boston_data$dis, boston_data$nox, cv = TRUE)
```

```
## Warning in smooth.spline(boston_data$dis, boston_data$nox, cv = TRUE): cross-
## validation with non-unique 'x' values seems doubtful
```

```
fit$df # best df using LOOCV
```

```
## [1] 15.42984
```

```
plot(boston_data$dis, boston_data$nox, xlim = dislims, cex = .5, col = "darkgrey")
lines(dis.grid, pred$fit, lwd = 2)
lines(dis.grid, pred$fit + 2 * pred$se, lty = "dashed")
lines(dis.grid, pred$fit - 2 * pred$se, lty = "dashed")
```


0.4
0.5
0.6
0.7
0.8
boston_data$nox
2
4
6
8
10
12
boston_data$dis

Question 2

- **(a)**

```r
weekly_data <- Weekly

fit <- glm(Direction ~ Lag1 + Lag2, data = weekly_data, family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = weekly_data)
##
## Deviance Residuals:
##    Min       1Q  Median       3Q      Max
## -1.623  -1.261   1.001   1.083   1.506
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22122    0.06147   3.599 0.000319 ***
## Lag1        -0.03872    0.02622  -1.477 0.139672
## Lag2         0.06025    0.02655   2.270 0.023232 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1488.2  on 1086  degrees of freedom
## AIC: 1494.2
##
## Number of Fisher Scoring iterations: 4
```

- **(b)**

```r
fit <- glm(Direction ~ Lag1 + Lag2, data = weekly_data[-1,], family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2, family = binomial, data = weekly_data[-1,
##     ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6258  -1.2617   0.9999   1.0819   1.5071
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.22324    0.06150   3.630 0.000283 ***
## Lag1        -0.03843    0.02622  -1.466 0.142683
## Lag2         0.06085    0.02656   2.291 0.021971 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1494.6  on 1087  degrees of freedom
## Residual deviance: 1486.5  on 1085  degrees of freedom
## AIC: 1492.5
##
## Number of Fisher Scoring iterations: 4
```

- **(c)**

```r
pred <- predict(fit, newdata = weekly_data[1,], type = 'response') # get predicted probability
pred
```

```
##         1
## 0.5713923
```

```r
weekly_data[1, 'Direction'] # get true label
```

```
## [1] Down
## Levels: Down Up
```

The predicted probability is .5713923, which is > 0.5, therefore the model incorrectly predicted up because the ground truth was down.

- **(d)**

```r
cv.error <- c()

for (i in 1:nrow(Weekly)) {
  fit <- glm(Direction ~ Lag1 + Lag2, data = Weekly[-i, ], family = "binomial") # fit excluding ith

  # if prob response is > .5 then Up is predicted
  # otherwise down is predicted
  prob <- predict(fit, newdata = Weekly[i, ], type = "response")
  pred <- ifelse(prob > 0.5, "Up", "Down")


  # if prediction is correct, error is 0
  # error is 1 for incorrect prediction
  cv.error[i] <- ifelse(pred != Weekly[i, "Direction"], 1, 0)
}
```

- **(e)**

```r
mean(cv.error) # error rate for model
```

```
## [1] 0.4499541
```

```r
mean(c(ifelse(weekly_data[,'Direction'] == 'Up', 1, 0))) # percentage of weeks that were up
```

```
## [1] 0.5555556
```

The error rate is ~.45 which is only slightly better than chance. Since the market has had a general upward trend over the years, we can see that just guessing up every single week would have given a ~55.5% chance of being correct, a 44.5% error rate, which is almost the exact same as our model. We would hope to be able to beat that 55.5% mark with a better model.
