GAM-HW

Franky Zhang

2/14/2022

7.3

Suppose we fit a curve with basis functions $b_1(X) = X$, $b_2(X) = (X-1)^2 I(X \geq 1)$. We fit linear regression model

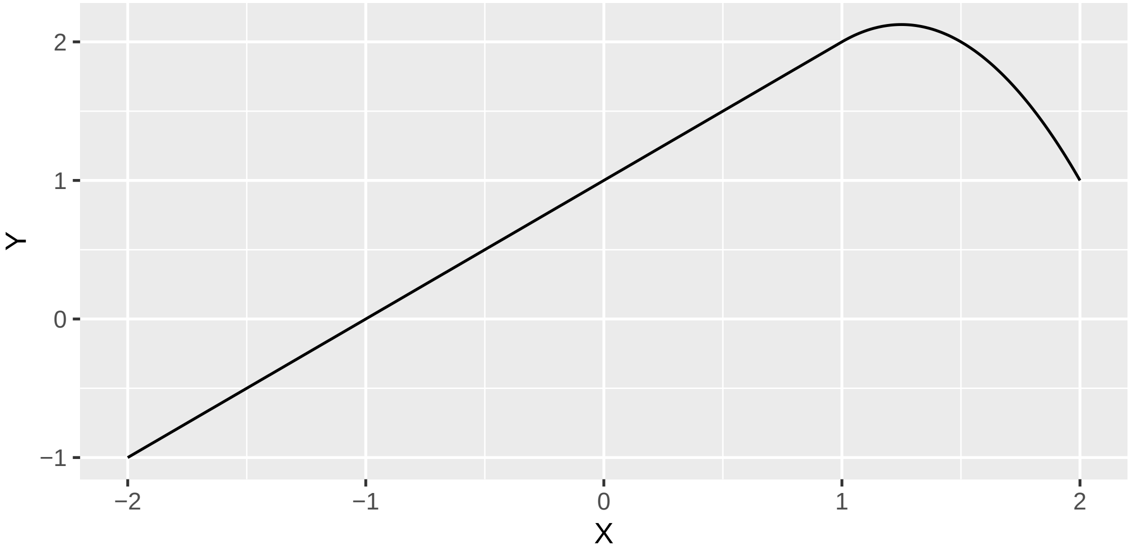
$$Y = \beta_0 + \beta_1 b_1(X) + \beta_2 b_2(X) + \epsilon$$

and obtain coefficient estimates $\hat{\beta}_0 = 1$, $\hat{\beta}_1 = 1$, $\hat{\beta}_2 = 3$. Sketch the estimated curve between $X = -2$ and $X = 2$. Note this intercepts, slopes and other relevant information.

Answer we can obtain piecewise regression line with the following function:

$$Y = \begin{cases} 1 + X & -2 \leq X < 1 \\ 1 + X - 2(X-1)^2 & 1 \leq X \leq 2 \end{cases}$$

```
X_1 <- seq(from = -2, to = 1, by = .01); Y_1 <- X_1 + 1
X_2 <- seq(from =  1, to = 2, by = .01); Y_2 <- X_2 + 1 - 2 * (X_2 - 1) * (X_2 - 1)
X <- c(X_1, X_2); Y <- c(Y_1, Y_2); rm(X_1, X_2, Y_1, Y_2)
ggplot() +
  geom_line(mapping = aes(x = X, y = Y))
```



```
rm(X, Y)
```

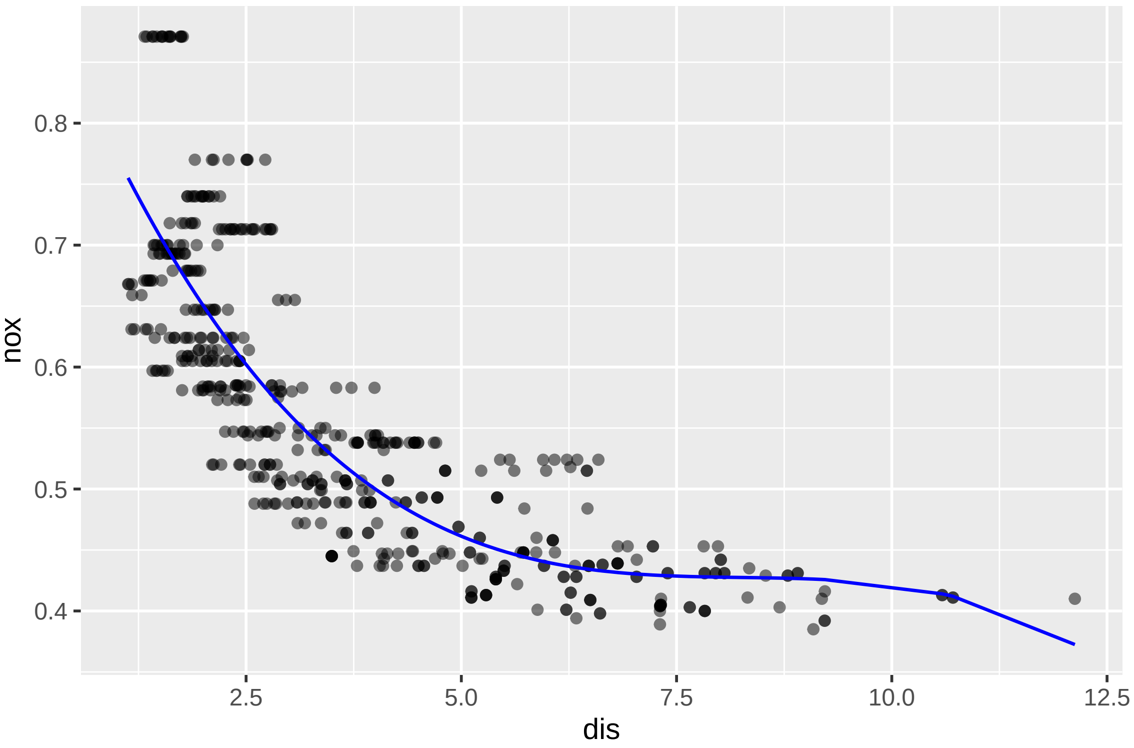
7.9

(a)

```
fit.1 <-lm(nox ~ poly(dis, 3), data = Boston)
summary(fit.1)$coef
```

```
##                 Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)    0.5546951 0.00275939 201.020894  0.000000e+00
## poly(dis, 3)1 -2.0030959 0.06207094 -32.271071 1.597201e-124
## poly(dis, 3)2  0.8563300 0.06207094  13.795987  6.133104e-37
## poly(dis, 3)3 -0.3180490 0.06207094  -5.123959  4.274950e-07
```

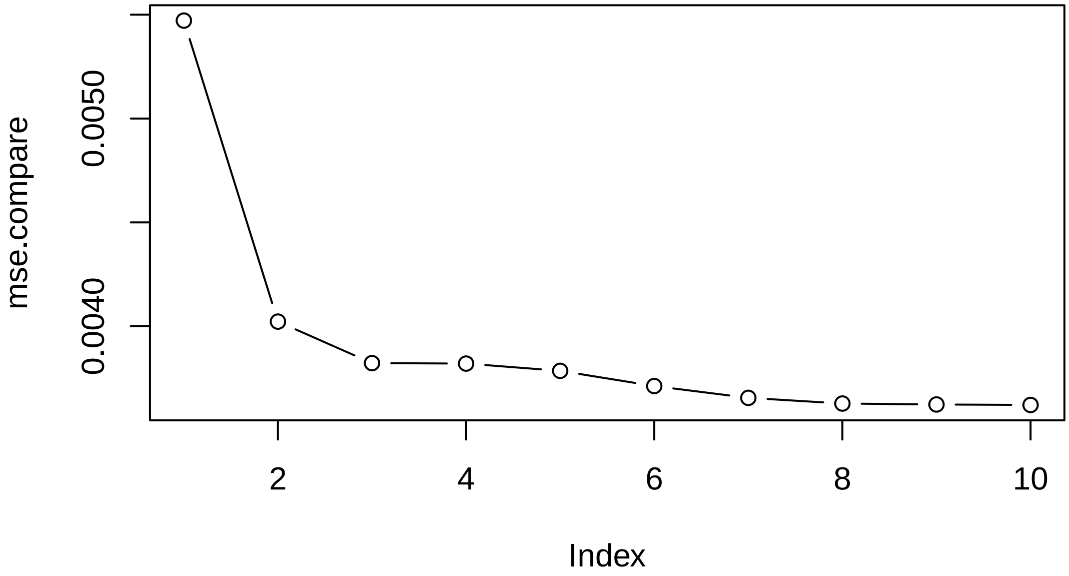
```
pred.1 <- predict(fit.1, newdata = Boston)
plot.data1 <- data.frame(dis  = Boston$dis,
                         nox  = Boston$nox,
                         pred = pred.1)
ggplot(data = plot.data1) +
  geom_point(aes(x = dis, y = nox), alpha = .5) +
  geom_line(aes(x = dis, y = pred), color = "blue", lwd = .6)
```



(b)

```
pred.compare <- data.frame(actual = Boston$nox)
for (i in 1:10) {
  fit <- lm(nox ~ poly(dis, i), data = Boston)
```

```r
  name <- paste("power", seq = "=", i)
  pred <- predict(fit, newdata = Boston)
  pred.compare <- cbind(pred.compare, pred)
  colnames(pred.compare)[i+1] <- name
  rm(fit, name, pred)
}
mse.compare <- c()
for (i in 1:10) {
  MSE <- sum((pred.compare$actual - pred.compare[, i+1])^2) / length(Boston$nox)
  mse.compare[i] <- MSE
  rm(MSE)
}
plot(mse.compare, type = "b")
```

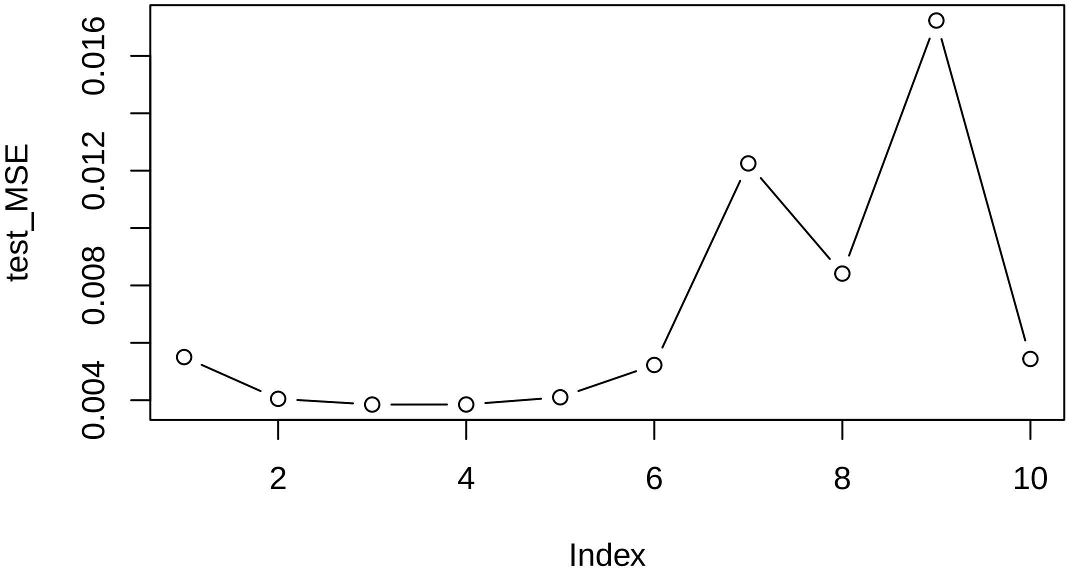


Answer:

The MSE continue to decrease as the increase of power.

(c)

```r
# perform cross-validation
k <- 10
set.seed(1147)
folds <- sample(rep(1:k, length = length(Boston$nox)))
# table(folds)
data <- data.frame(nox = Boston$nox, dis = Boston$dis, folds = folds)
cv.errors <- data.frame()
for (j in 1:k) {
  train_data <- data[folds != j, ]
  test_data  <- data[folds == j, ]
  for (i in 1:10) {
    fit <- lm(nox ~ poly(dis, i), data = train_data)
    col.name <- paste("power", seq = "=", i)
    row.name <- paste("fold" , seq = "=", j)
    pred  <- predict(fit, newdata = test_data)
    MSE <- (sum((test_data$nox - pred)^2))/length(test_data$nox)
    cv.errors[j, i] <- MSE
    colnames(cv.errors)[i] <- col.name
    rownames(cv.errors)[j] <- row.name
    rm(fit, col.name, row.name, pred, MSE)
  }
  rm(train_data, test_data)
}
test_MSE <- apply(cv.errors, 2, mean)
plot(test_MSE, type = "b")
```



```
which.min(apply(cv.errors, 2, mean))
```

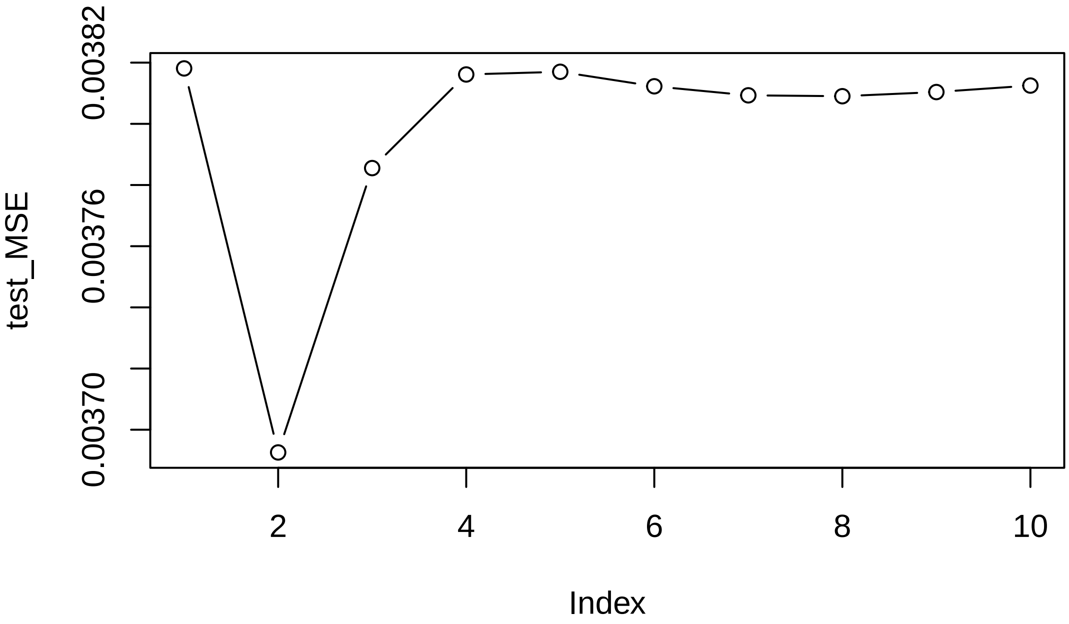
```
## power = 3
##         3
```

Answer:

As to my consideration, the optimal power for the polynomial regression is 3. According to the plot, when power = 3, the test error reached the minimum.

(d)

```
# fit a regression spline line
cv.errors <- data.frame()
for (j in 1:k) {
  train_data <- data[folds != j, ]
  test_data  <- data[folds == j, ]
  for (i in 1:10) {
    fit <- lm(nox ~ bs(dis, knots = i, df = 4), data = data)
    col.name <- paste("knots", seq = "=", i)
    row.name <- paste("fold" , seq = "=", j)
    pred  <- predict(fit, newdata = test_data)
    MSE <- (sum((test_data$nox - pred)^2))/length(test_data$nox)
    cv.errors[j, i] <- MSE
    colnames(cv.errors)[i] <- col.name
    rownames(cv.errors)[j] <- row.name
    rm(fit, col.name, row.name, pred, MSE)
  }
  rm(train_data, test_data)
}
test_MSE <- apply(cv.errors, 2, mean)
plot(test_MSE, type = "b")
```

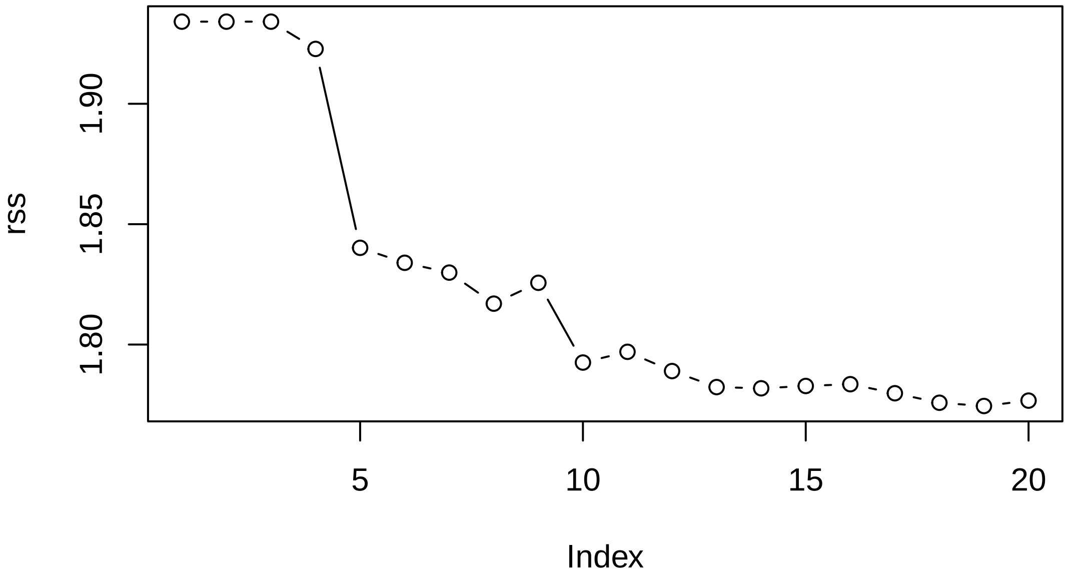


Answer:

Utilize cross validation to find that the optimal number of knots is 2

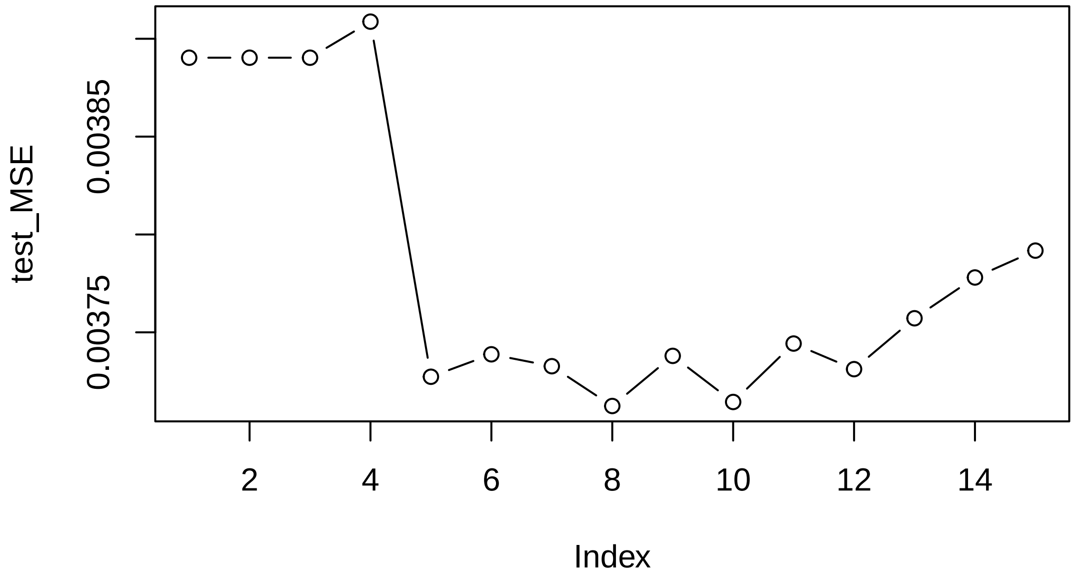
(e)

```
# fit a regression spline line
rss <- c()
for (j in 1:k) {
  train_data <- data[folds != j, ]
  test_data  <- data[folds == j, ]
  for (i in 1:20) {
    fit <- lm(nox ~ bs(dis, df = i), data = data)
    rss[i] <- sum((fit$residuals)^2)
  }
  rm(train_data, test_data)
}
plot(rss, type = "b")
```



(f)

```
k <- 10
set.seed(1518)
folds <- sample(rep(1:k, length = length(Boston$nox)))
# table(folds)
data <- data.frame(nox = Boston$nox, dis = Boston$dis, folds = folds)
cv.errors <- data.frame()
for (j in 1:k) {
  train_data <- data[folds != j, ]
  test_data  <- data[folds == j, ]
  for (i in 1:15) {
    fit <- lm(nox ~ bs(dis, df = i), data = train_data)
    col.name <- paste("df", seq = "=", i)
    row.name <- paste("fold" , seq = "=", j)
    pred  <- predict(fit, newdata = test_data)
    MSE <- (sum((test_data$nox - pred)^2))/length(test_data$nox)
    cv.errors[j, i] <- MSE
    colnames(cv.errors)[i] <- col.name
    rownames(cv.errors)[j] <- row.name
    rm(fit, col.name, row.name, pred, MSE)
  }
  rm(train_data, test_data)
}
test_MSE <- apply(cv.errors, 2, mean)
plot(test_MSE, type = "b")
```



```
rm(test_MSE, fit.1, pred.1, cv.errors, fit.bs,
   plot.data1, data, pred.compare,
   i, j, k, folds, mse.compare, rss)
```

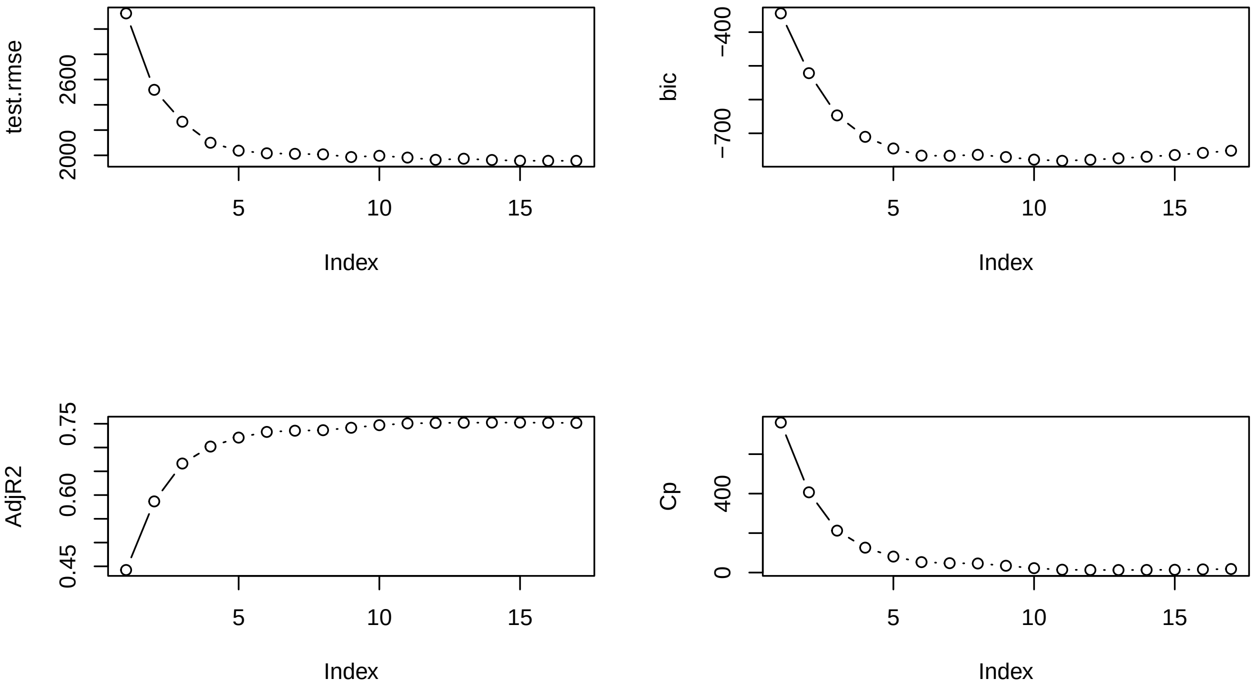
Answer:

I use cross validation to find the optimal freedom of degree is 8.

7.10

(a)

```r
set.seed(1543)
train <- sample(c(TRUE, FALSE), dim(College)[1], replace = TRUE, prob = c(.8, .2))
training <- College[train, ]
test <- College[!train, ]
fit.forward <- regsubsets(Outstate~., data = training, nvmax = 17, method = "forward")
fit.summary <- summary(fit.forward)
test.mat <- model.matrix(Outstate~., data = test)
test.rmse <- c()
bic <- fit.summary$bic; AdjR2 <- fit.summary$adjr2; Cp <- fit.summary$cp
for (i in 1:17) {
  coefi <- coef(fit.forward, id = i)
  pred <- test.mat[, names(coefi)] %*% coefi
  test.rmse[i] <- sqrt(sum((pred - test$Outstate)^2) / length(test$Outstate))
  rm(coefi, pred)
}
par(mfrow = c(2, 2))
plot(test.rmse, type = "b")
plot(bic, type = "b")
plot(AdjR2, type = "b")
plot(Cp, type = "b")
```



Answer:

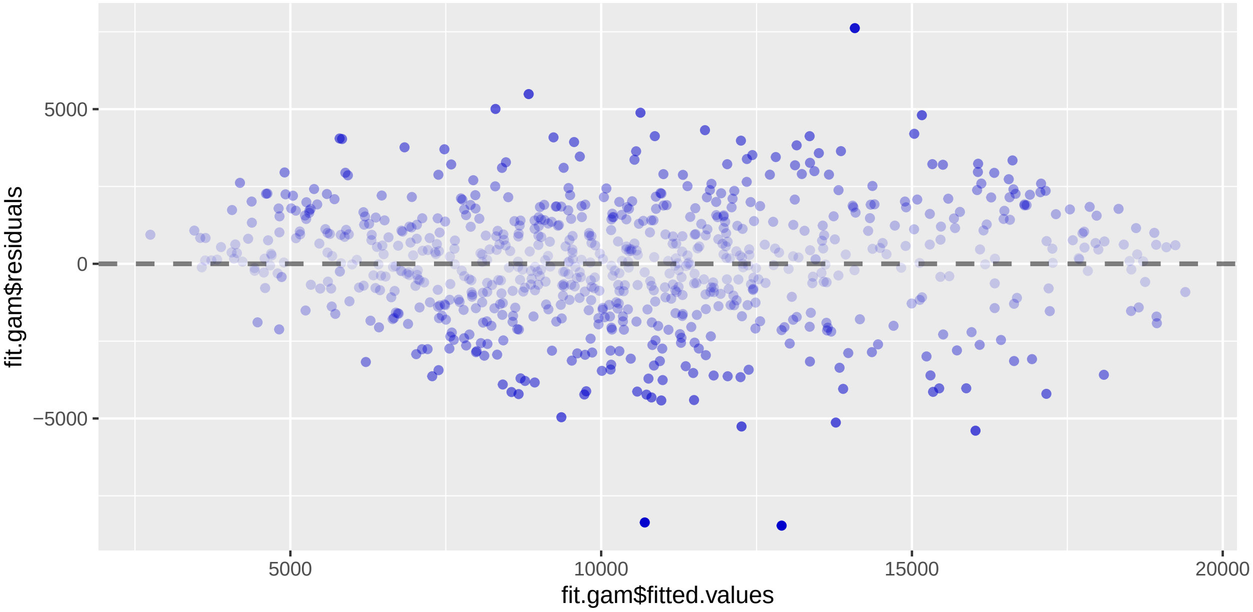
According to the aforementioned plot, I finally choose five predictor to continue the following study. The reason is that test MSE decreased rapidly before #predictors ≤ 5. As the number keeped increasing, there was no clear evidence showing that more complex model is better. Therefore, 5 is the optimal number to keep the balance of prediciton accuracy and model interpretability.

(b)

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Private + s(Room.Board, 1) + PhD + perc.alumni + Expend
## Model 2: Outstate ~ Private + s(Room.Board, 2) + PhD + perc.alumni + Expend
## Model 3: Outstate ~ Private + s(Room.Board, 3) + PhD + perc.alumni + Expend
## Model 4: Outstate ~ Private + s(Room.Board, 4) + PhD + perc.alumni + Expend
## Model 5: Outstate ~ Private + s(Room.Board, 5) + PhD + perc.alumni + Expend
## Model 6: Outstate ~ Private + s(Room.Board, 6) + PhD + perc.alumni + Expend
##   Resid. Df Resid. Dev      Df Deviance Pr(>Chi)
## 1       771 3302765962
## 2       770 3279785901 0.99998 22980061  0.01962 *
## 3       769 3267956018 1.00002 11829883  0.09408 .
## 4       768 3255702066 0.99997 12253952  0.08837 .
## 5       767 3243194276 1.00024 12507791  0.08517 .
## 6       766 3232551183 0.99985 10643093  0.11224
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 1) + perc.alumni +
##     Expend
## Model 2: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 2) + perc.alumni +
##     Expend
## Model 3: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     Expend
## Model 4: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 4) + perc.alumni +
##     Expend
## Model 5: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 5) + perc.alumni +
##     Expend
## Model 6: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 6) + perc.alumni +
##     Expend
##   Resid. Df Resid. Dev      Df Deviance  Pr(>Chi)
## 1       770 3279785569
## 2       769 3222696619 0.99987 57088950 0.0002139 ***
## 3       768 3204928321 0.99987 17768298 0.0388890 *
## 4       767 3196052070 1.00020  8876251 0.1444018
## 5       766 3190595970 1.00007  5456100 0.2524567
## 6       765 3186753772 1.00034  3842198 0.3369758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Analysis of Deviance Table
##
## Model 1: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 1)
## Model 2: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 2)
## Model 3: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 3)
## Model 4: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 4)
## Model 5: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 5)
## Model 6: Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni +
##     s(Expend, 6)
##   Resid. Df Resid. Dev      Df  Deviance  Pr(>Chi)
## 1       768 3204834579
## 2       767 2926539833 0.99978 278294746 < 2.2e-16 ***
## 3       766 2854448084 1.00006  72091749 9.098e-06 ***
## 4       765 2824987529 0.99987  29460554  0.004556 **
## 5       764 2805369518 1.00030  19618011  0.020628 *
## 6       763 2793285879 1.00009  12083639  0.069259 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Answer: The final model I choose is 'fit.2.3.5 <- gam(Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni + s(Expend, 5), data = data)' via analysis of covariance.

(c)

```
fit.gam <- gam(Outstate ~ Private + s(Room.Board, 2) + s(PhD, 3) + perc.alumni
               + s(Expend, 5), data = data[train, ])
postResample(predict(fit.gam, data[!train, ]), data[!train, ]$Outstate)
```

```
##         RMSE     Rsquared          MAE
## 1790.9293573    0.8296377 1409.3115544
```

(d)

```
fit.lm <- lm(Outstate~., data = data[train, ])
postResample(predict(fit.lm, data[!train, ]), data[!train, ]$Outstate)
```

```
##          RMSE     Rsquared          MAE
## 2036.9101864    0.7779529 1604.6945412
```

```
postResample(predict(fit.gam, data[!train, ]), data[!train, ]$Outstate)
```

```
##          RMSE     Rsquared          MAE
## 1790.9293573    0.8296377 1409.3115544
```

7.11

(a)

```
set.seed(1044)
Y  <- rnorm(100, mean = 10)
X1 <- rnorm(100, mean = 8)
X2 <- rnorm(100, mean = 6)
```

(b)

```
set.seed(1047)
beta1 <- rnorm(n = 1, mean = 3, sd = 1)
```

(c)

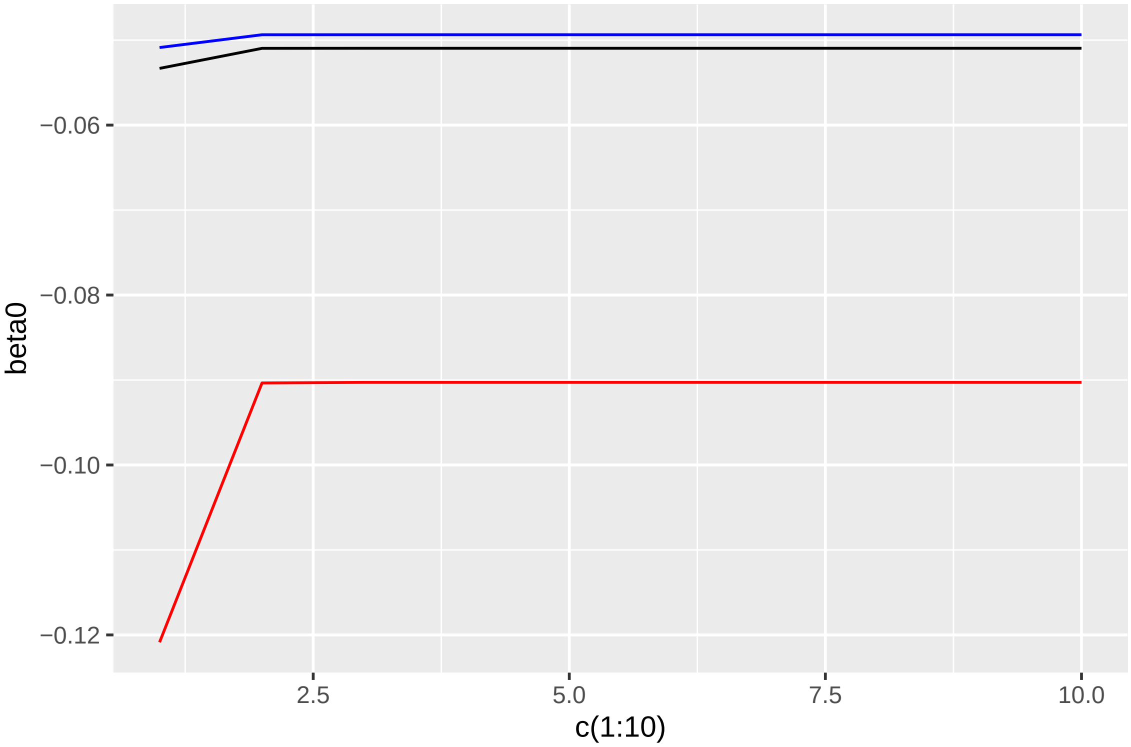
```
a <- Y - beta1 * X1
beta2 <- lm(a ~ X2)$coef[2]
```

(d)

```
a <- Y - beta2 * X2
beta1 <- lm(a ~ X1)$coef[2]
```

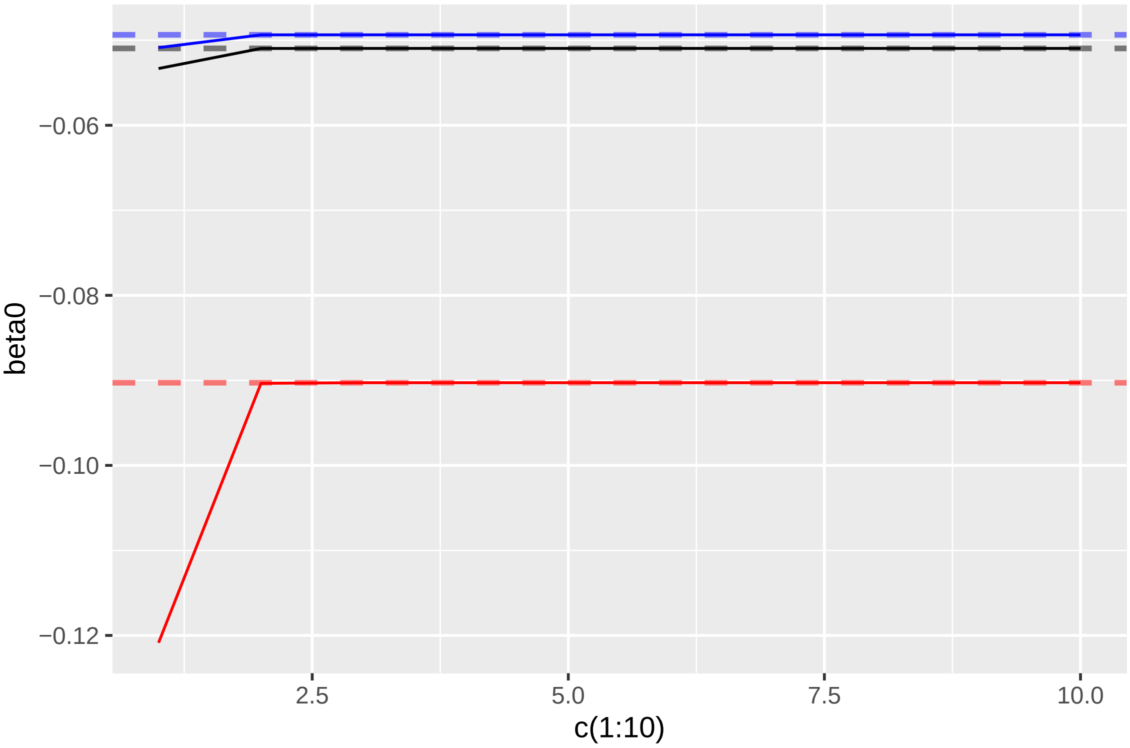
(e)

```
set.seed(1044)
Y  <- rnorm(100)
X1 <- rnorm(100)
X2 <- rnorm(100)
set.seed(1053)
beta1 <- rnorm(n = 1, mean = 0, sd = 1)
# for loop
record <- data.frame()
for (i in 1:10) {
  a <- Y - beta1 * X1
  beta2 <- lm(a ~ X2)$coef[2]
  a <- Y - beta2 * X2
  beta1 <- lm(a ~ X1)$coef[2]
  beta0 <- lm(a ~ X1)$coef[1]
  # print(beta0)
  # print(beta1)
  # print(beta2)
  record <- rbind(record, c(beta0, beta1, beta2))
}
colnames(record) <- c("beta0", "beta1", "beta2")
ggplot(data = record) +
  geom_line(aes(c(1:10), beta0), color = "black", lwd = .5) +
  geom_line(aes(c(1:10), beta1), color = "blue", lwd = .5) +
  geom_line(aes(c(1:10), beta2), color = "red", lwd = .5)
```



(f)

```
fit.multi <- lm(Y~X1+X2)
ggplot(data = record) +
  geom_line(aes(c(1:10), beta0), color = "black", lwd = .5) +
  geom_hline(yintercept = fit.multi$coefficients[1], color = "black",
             linetype = 2, lwd = 1, alpha = .5) +
  geom_line(aes(c(1:10), beta1), color = "blue", lwd = .5) +
  geom_hline(yintercept = fit.multi$coefficients[2], color = "blue",
             linetype = 2, lwd = 1, alpha = .5) +
  geom_line(aes(c(1:10), beta2), color = "red", lwd = .5) +
  geom_hline(yintercept = fit.multi$coefficients[3], color = "red",
             linetype = 2, lwd = 1, alpha = .5)
```



(g)

Answer:

under this case, 3 is a good interation number.