ch7-hw

Bruce Mallory

2/27/2021

7.3/ use $b_1(x) = x$ when $x < 1$

use $b_1(x)$ & $b_2(x)$ when $x \geq 1$

$b_2(x) = (x-1)^2$

$y = 1 + 1\,b_1(x) + -2\,b_2(x)$

$$y = \begin{cases} 1+1x, & x < 1 \\ 1 + 1x + -2(x-1)^2, & x \geq 1 \end{cases}$$

$1 + x - 2(x^2 - 2x + 1)$

$1 + x - 2x^2 + 4x - 2$

$-2x^2 + 5x - 1$

$-2(1)^2 + 5(1) - 1 = 2$

$-2(2)^2 + 5(2) - 1 = -8 + 10 - 1 = 1$

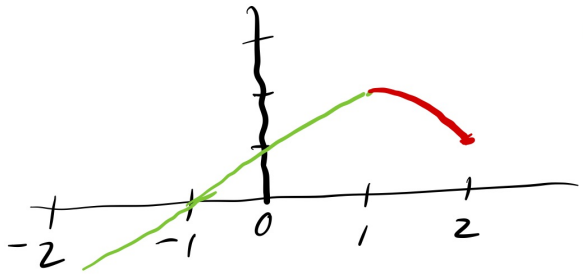
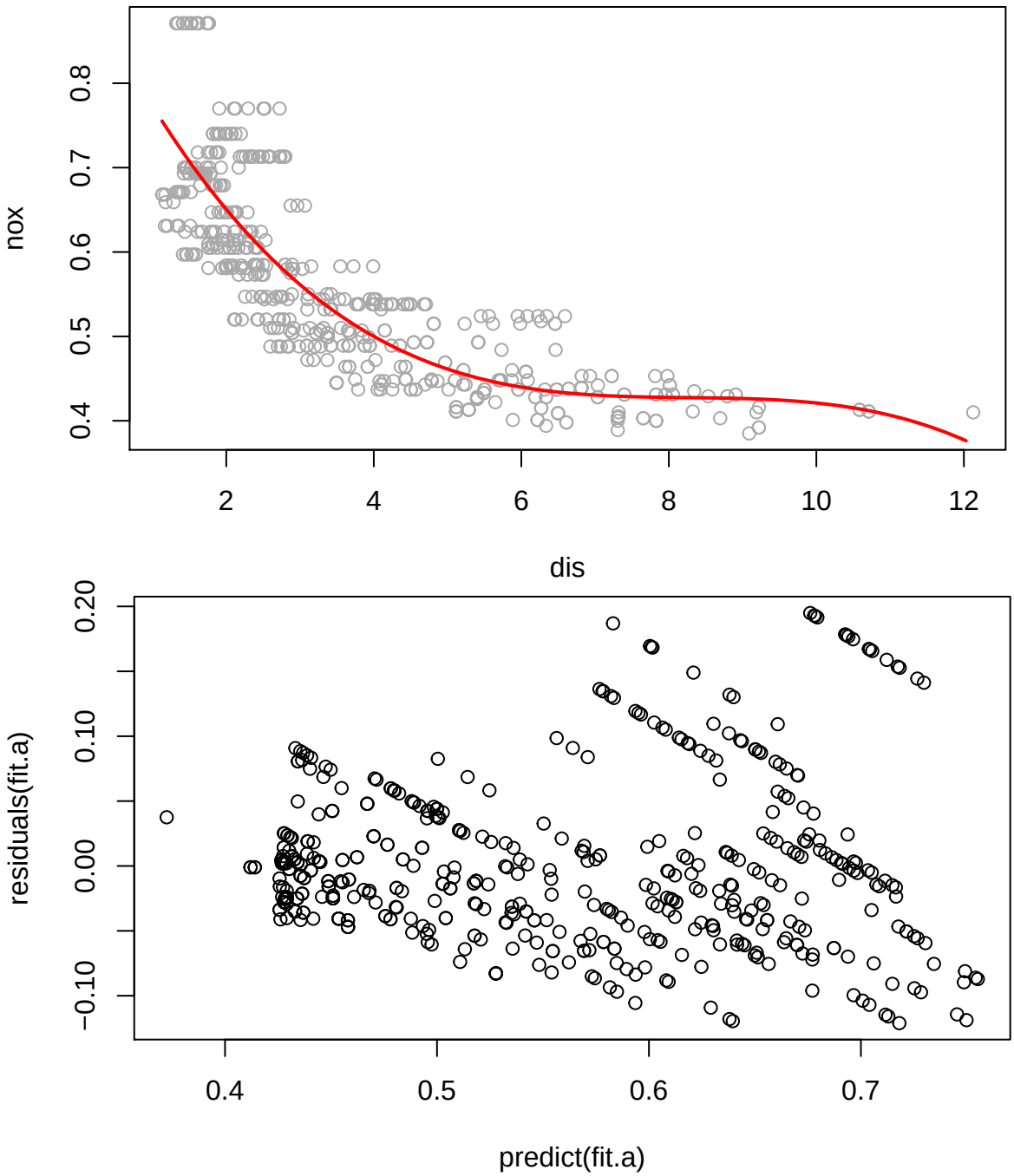
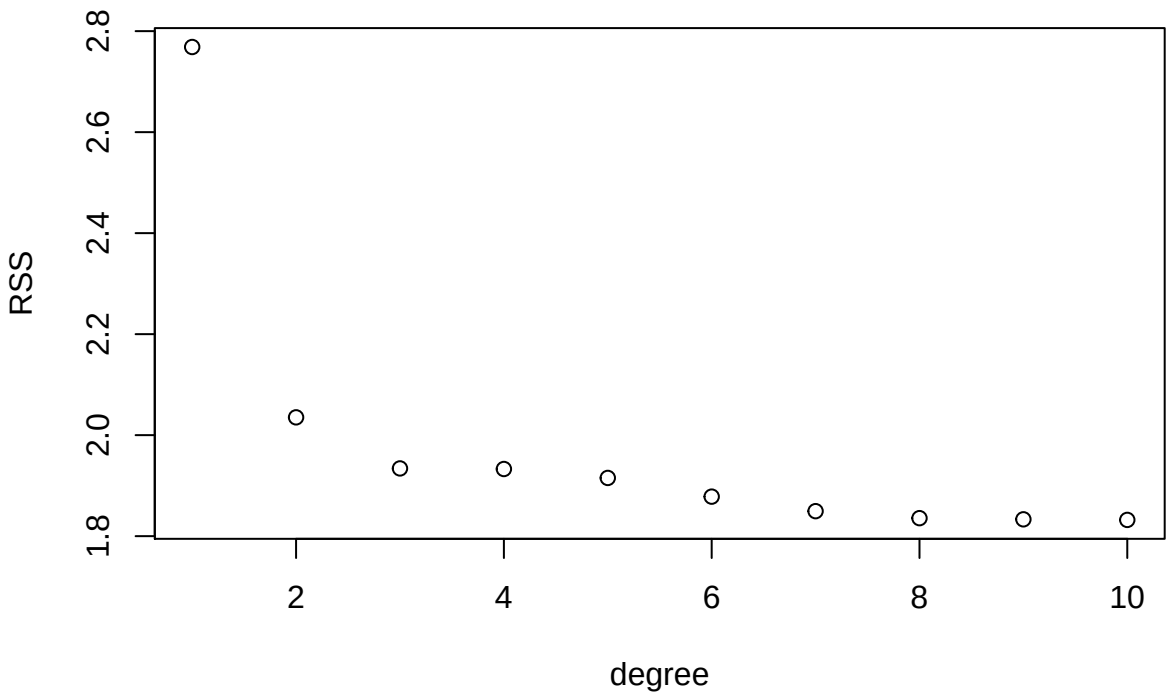


Figure 1: problem 7.3

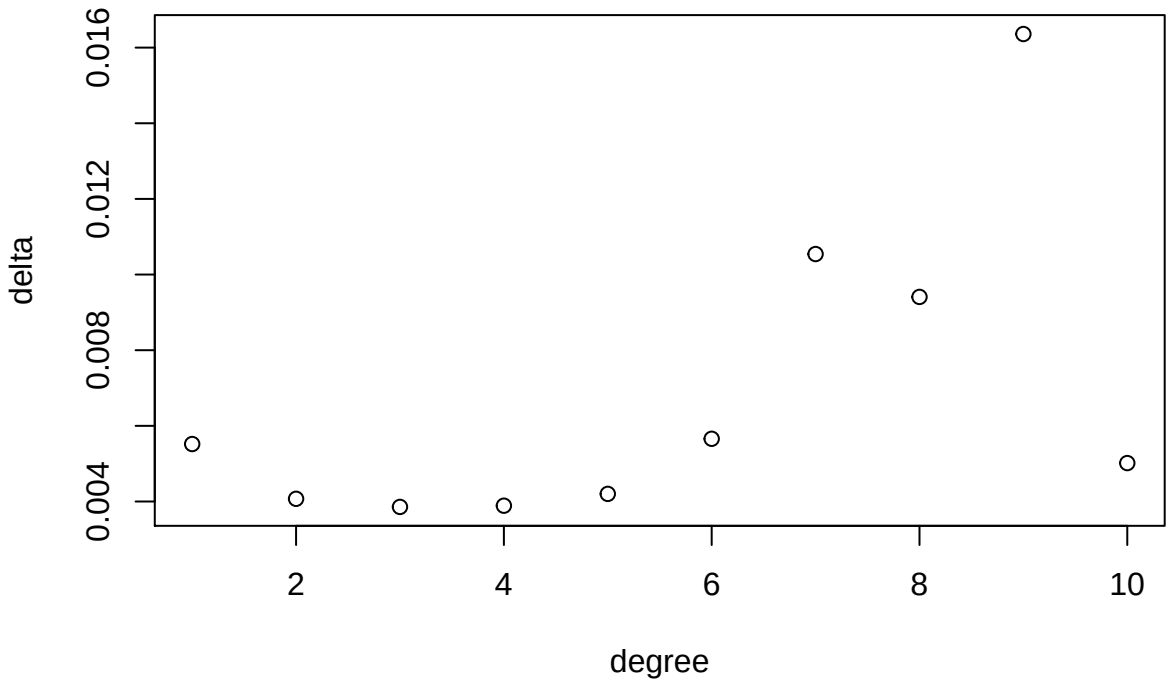
7.9




```
#b
degree <- rep(NA,10)
RSS <- rep(NA,10)
for (d in 1:10) {
  fit <- lm(nox~poly(dis,d),data=Boston)
  degree[d] <- d
  RSS[d] <- sum(fit$residuals^2)
}
plot(degree, RSS)
```



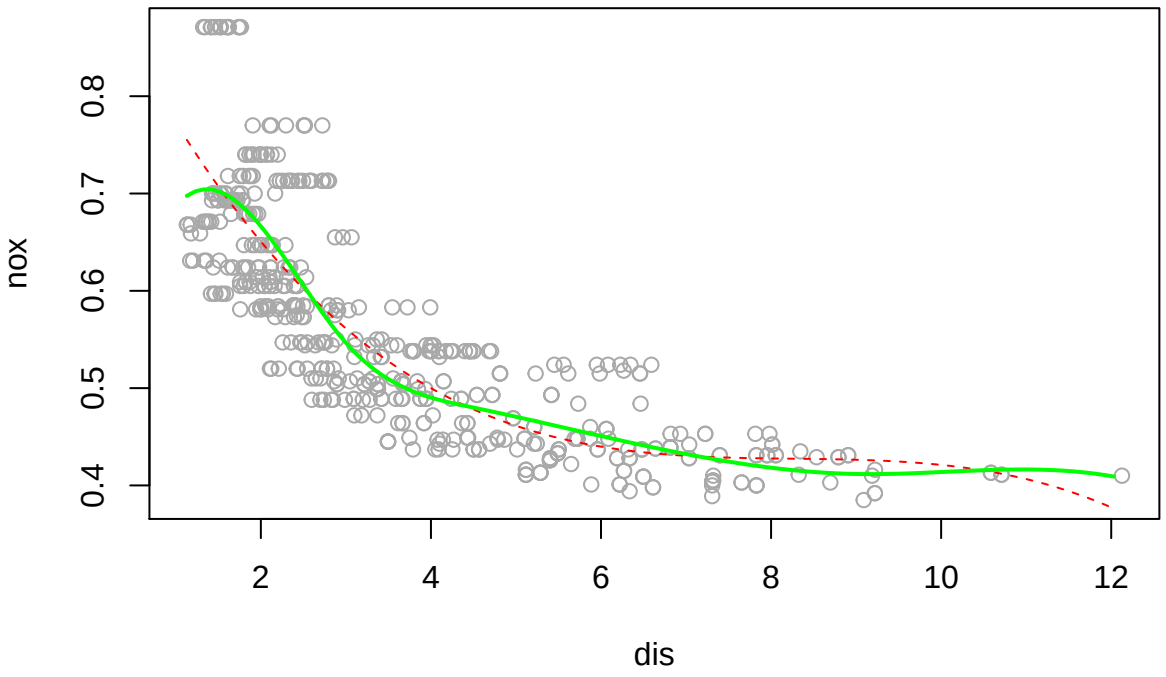

```r
#c
delta <- rep(NA,10)
for (d in 1:10) {
  fit <- glm(nox~poly(dis,d),data=Boston)
  degree[d] <- d
  delta[d] <- cv.glm(Boston, fit, K=10)$delta[2]
}
plot(degree, delta)
```



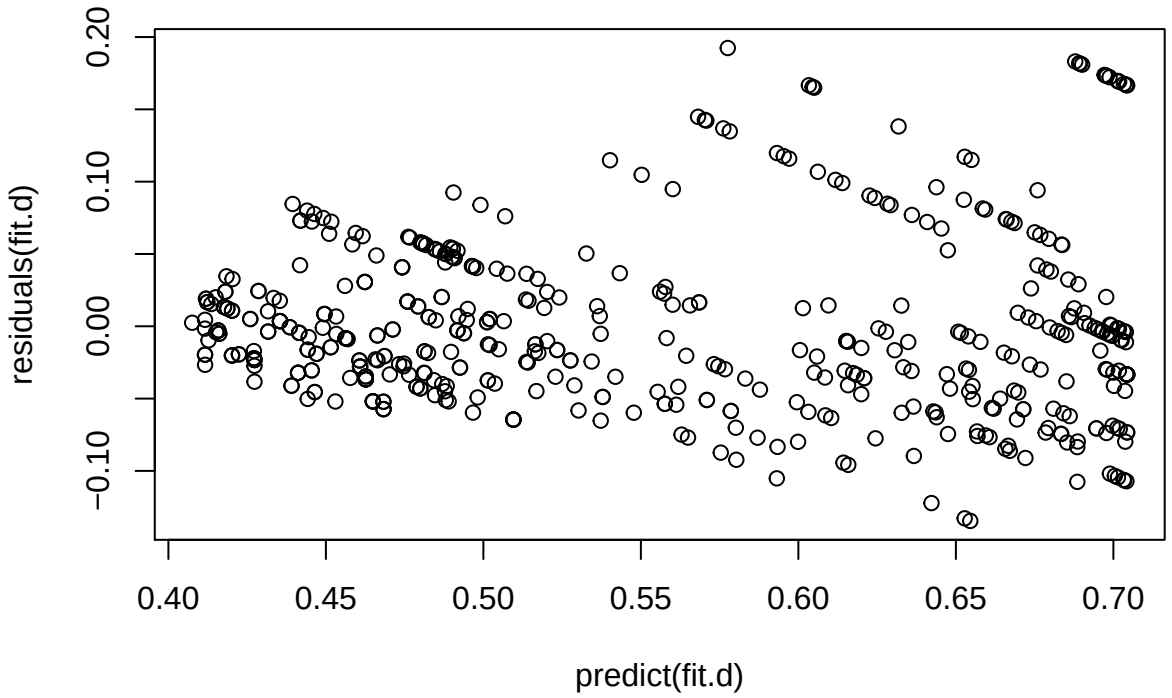
```
## The polynomial with lowest cross validation error is polynomial with degree 3
```

When this chunk is run several times, the cross validation error consistantly decreases until about degree 3 or 4, and then increases. An interesting artifact (of overfitting?) is that in about 1 in 10 runs of this chunk, the degree 10 polynomial has the smallest cross validation error.

```
#d
fit.d <- lm(nox~bs(dis, df=4, knots=c(3, 4.5, 9)), data=Boston)
#summary(fit.d)

dislim = range(Boston$dis)
dis.grid = seq(from=dislim[1], to=dislim[2], by=0.1)
fit.d.pred = predict(fit.d, list(dis=dis.grid))
plot(nox~dis, data=Boston, col="darkgrey")
lines(dis.grid, fit.a.pred, col="red", lwd=1, lty=2)
lines(dis.grid, fit.d.pred, col="green", lwd=2)
```

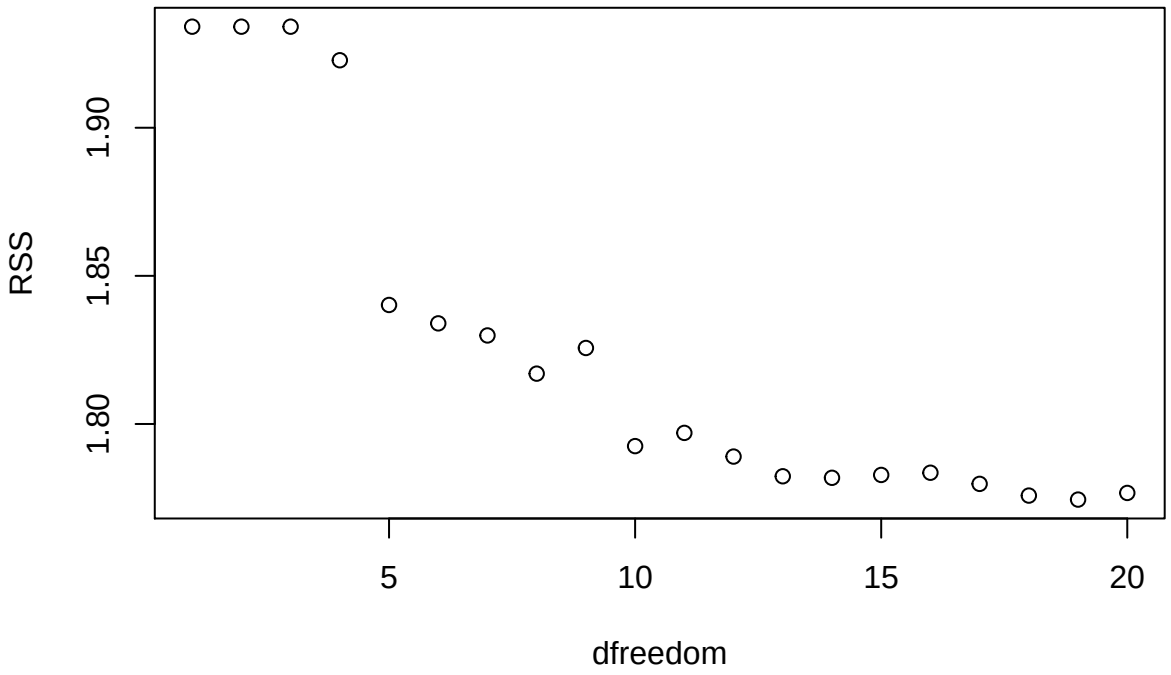



```
plot(predict(fit.d), residuals(fit.d))
```

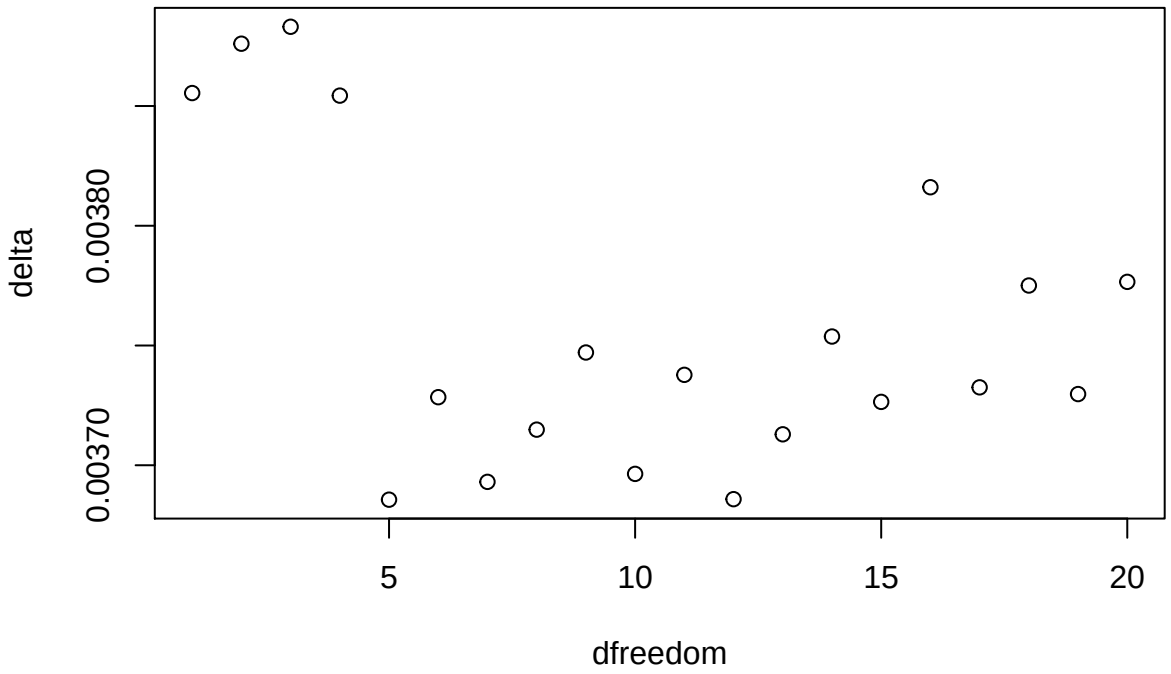


I chose the knots by looking at the dis vs. nox plot and noticing the places where it visually looked like nox was changing quickly (the horizontal gaps in the scatterplot). I've also plotted (red dotted line) the original poly fit from part a.

```
#e
dfreedom <- rep(NA,20)
RSS <- rep(NA,20)
for (d in 1:20) {
  fit <- lm(nox~bs(dis, df=d), data=Boston)
  dfreedom[d] <- d
  RSS[d] <- sum(fit$residuals^2)
}
plot(dfreedom, RSS)
```




```
#f
delta <- rep(NA,20)
for (d in 1:20) {
  fit <- glm(nox~bs(dis, df=d), data=Boston)
  dfreedom[d] <- d
  delta[d] <- cv.glm(Boston, fit, K=10)$delta[2]
}
plot(dfreedom, delta)
```




```
## The degrees of freedom for the regression splines with the lowest cross validation error is this many
```

Running this chunk many times, I've seen that the lowest cross validation error is more often than not 10 degrees of freedom.
7.10

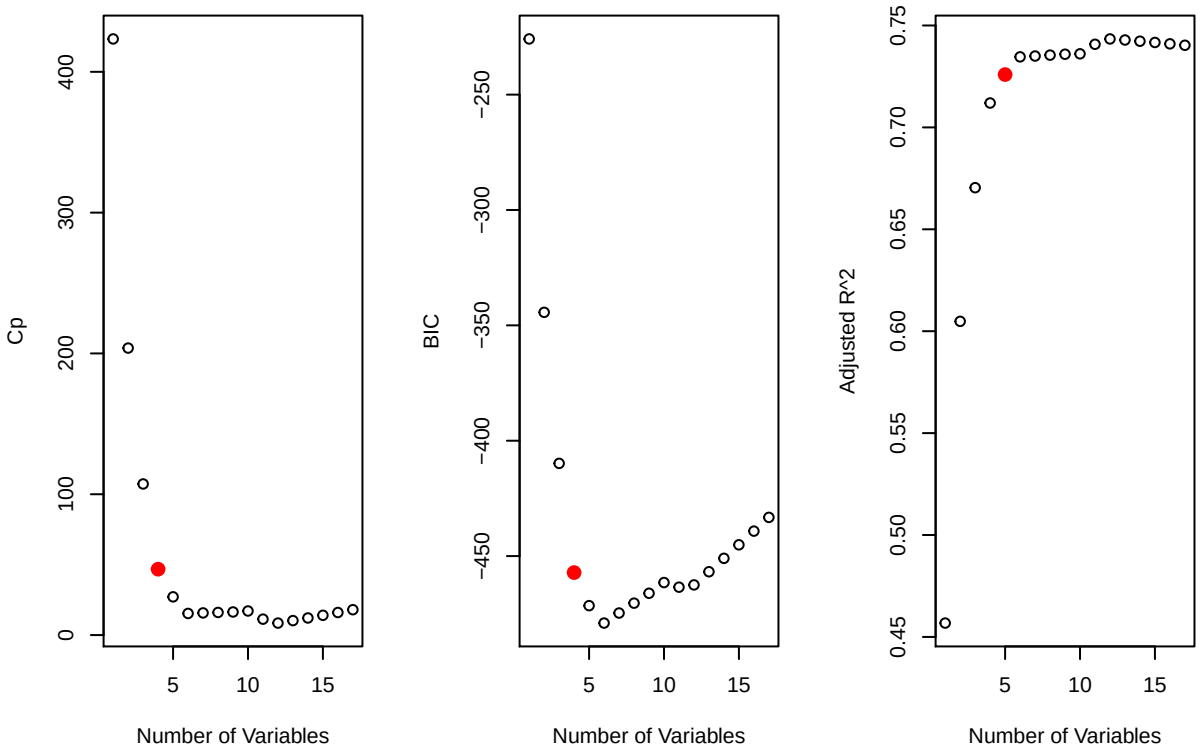
```
these <- sample(nrow(College),nrow(College)/2)
train <- College[these,]
test <- College[-these,]

#a
fit.a <- regsubsets(Outstate~.,data=train, nvmax=17, method="forward")
#summary(fit.a)
par(mfrow=c(1,3))
#Cp
plot(summary(fit.a)$cp,xlab="Number of Variables",ylab="Cp")
small.cp <- which.min(summary(fit.a)$cp)
cp.se <- sd(summary(fit.a)$cp)
for (i in 1:17){
  if (summary(fit.a)$cp[i] < (summary(fit.a)$cp[small.cp] + .2*cp.se)){
    break
  }
  smallest.cp <- i
}
points(smallest.cp, summary(fit.a)$cp[smallest.cp], col="red", cex=2, pch=20)
cat("The simplest model within .2 of a standard deviation of the the lowest Cp has this many variables:
```

```
## The simplest model within .2 of a standard deviation of the the lowest Cp has this many variables: 4
```

```
#BIC
plot(summary(fit.a)$bic,xlab="Number of Variables",ylab="BIC")
small.bic <- which.min(summary(fit.a)$bic)
bic.se <- sd(summary(fit.a)$bic)
for (i in 1:17){
  if (summary(fit.a)$bic[i] < (summary(fit.a)$bic[small.bic] + .2*bic.se)){
    break
  }
  smallest.bic <- i
}
points(smallest.bic, summary(fit.a)$bic[smallest.bic], col="red", cex=2, pch=20)
cat("The simplest model within .2 of a standard deviation of the the lowest BIC has this many variables
```

```
## The simplest model within .2 of a standard deviation of the the lowest BIC has this many variables: 4
```

```
#adjR2
plot(summary(fit.a)$adjr2,xlab="Number of Variables",ylab="Adjusted R^2")
big.adjr2 <- which.max(summary(fit.a)$adjr2)
adjr2.se <- sd(summary(fit.a)$adjr2)
for (i in 1:17){
  if (summary(fit.a)$adjr2[i] > (summary(fit.a)$adjr2[big.adjr2] - .2*adjr2.se)){
    break
  }
  smallest.adjr2 <- i
}
points(smallest.adjr2, summary(fit.a)$adjr2[smallest.adjr2], col="red", cex=2, pch=20)
```




```
cat("The simplest model within .2 of a standard deviation of the the lowest AdjustedR^2 has this many va
```

```
## The simplest model within .2 of a standard deviation of the the lowest AdjustedR^2 has this many vari
```

Looking at the Cp, the BIC, and the Adjusted R^2 and looking for the simpliest model close to the lowest (highest for Adjusted R^2) I've decided to use 5 variables in my model.

```
#b
```