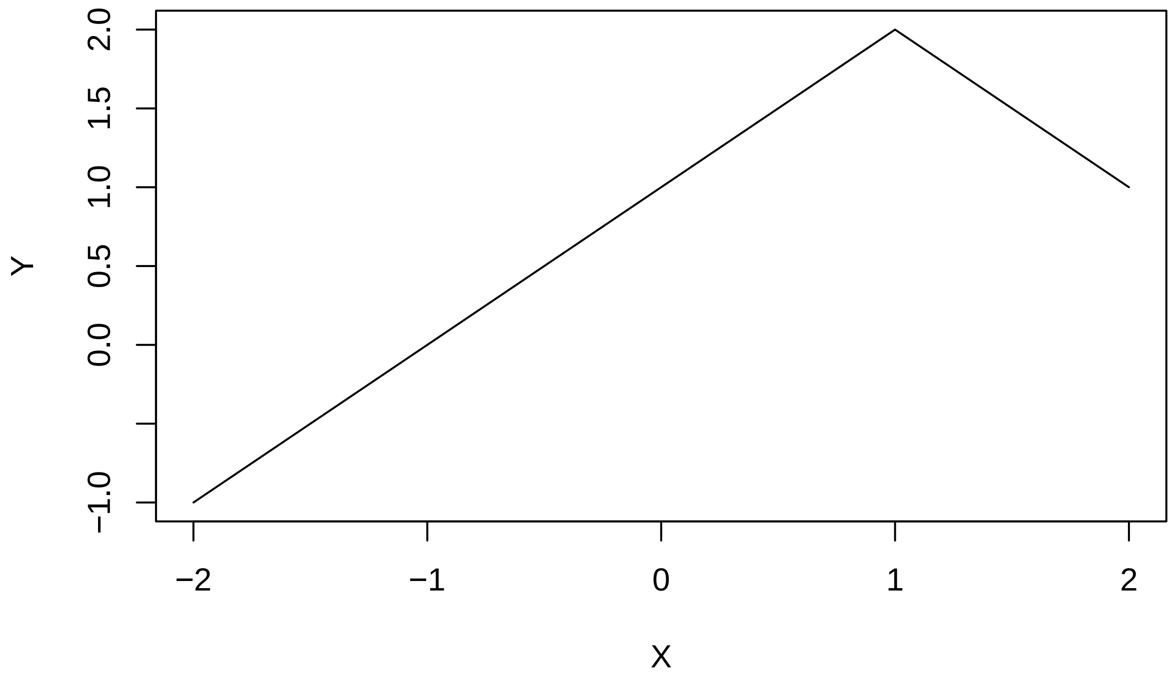
GAM Homework

Pruthvi Bharadwaj

Mar 4, 2021

3.

```r
X <- -2:2
Y <- 1 + 1*X - 2*((X - 1)^2)*I(X >= 1)
plot(X, Y, type = "l")
```



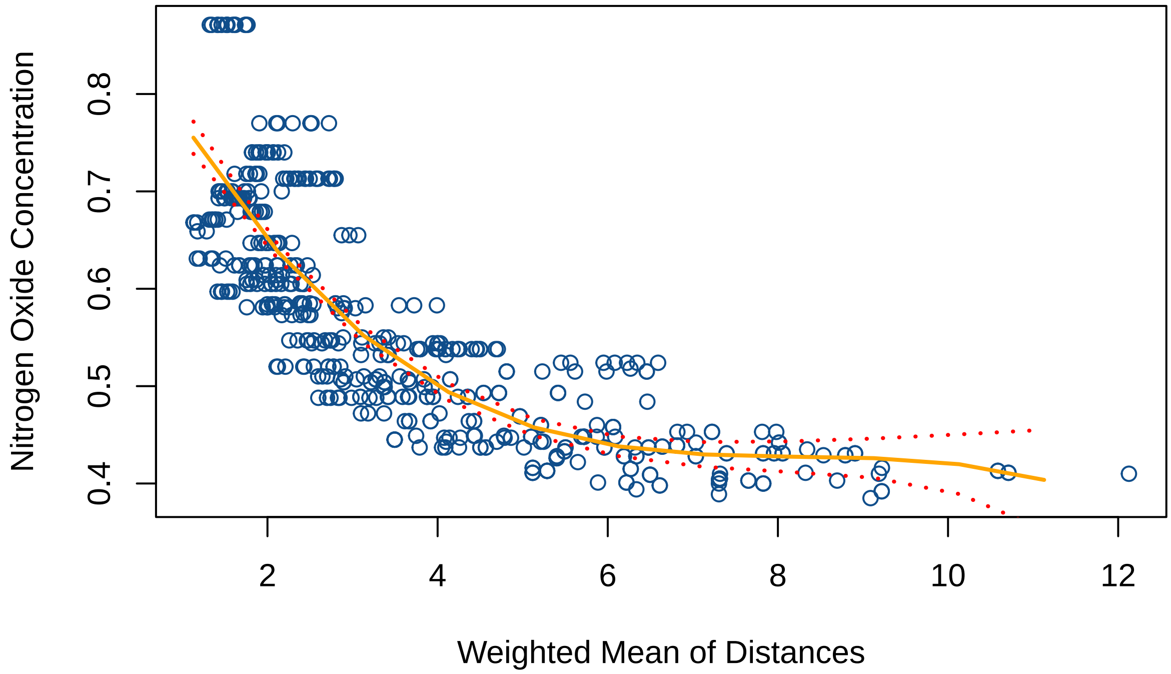
9.(a)

```r
set.seed(1)
fit7.9a <- lm(nox ~ poly(dis, 3), data = Boston)
summary(fit7.9a)
```

```
##
```

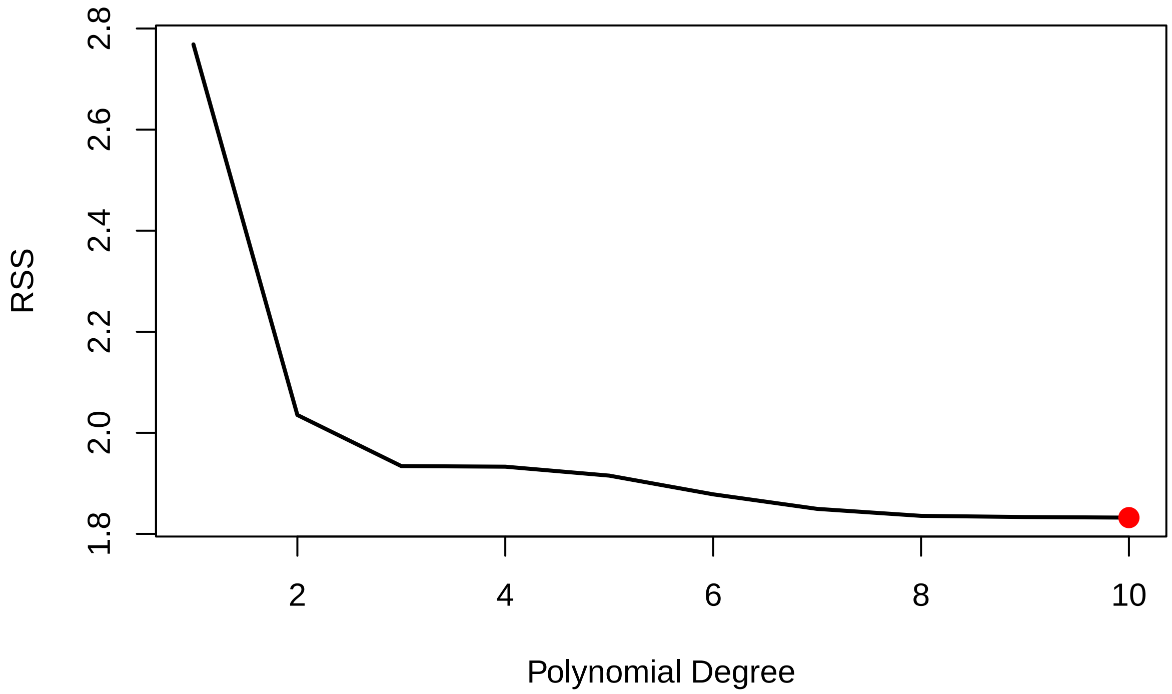
```
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 201.021  < 2e-16 ***
## poly(dis, 3)1 -2.003096   0.062071 -32.271  < 2e-16 ***
## poly(dis, 3)2  0.856330   0.062071  13.796  < 2e-16 ***
## poly(dis, 3)3 -0.318049   0.062071  -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
lims_dis <- range(Boston$dis)
grid_dis <- seq(lims_dis[1], lims_dis[2])
pred1 <- predict(fit7.9a, list(dis = grid_dis), se = TRUE)
se_lines <- cbind(pred1$fit + 2*pred1$se.fit, pred1$fit - 2*pred1$se.fit)
plot(Boston$dis, Boston$nox, xlab = "Weighted Mean of Distances", ylab = "Nitrogen Oxide Concentration"
lines(grid_dis, pred1$fit, col = "orange", lwd = 2)
matlines(grid_dis, se_lines, lwd = 2, col = "red", lty = 3)
```



(b)

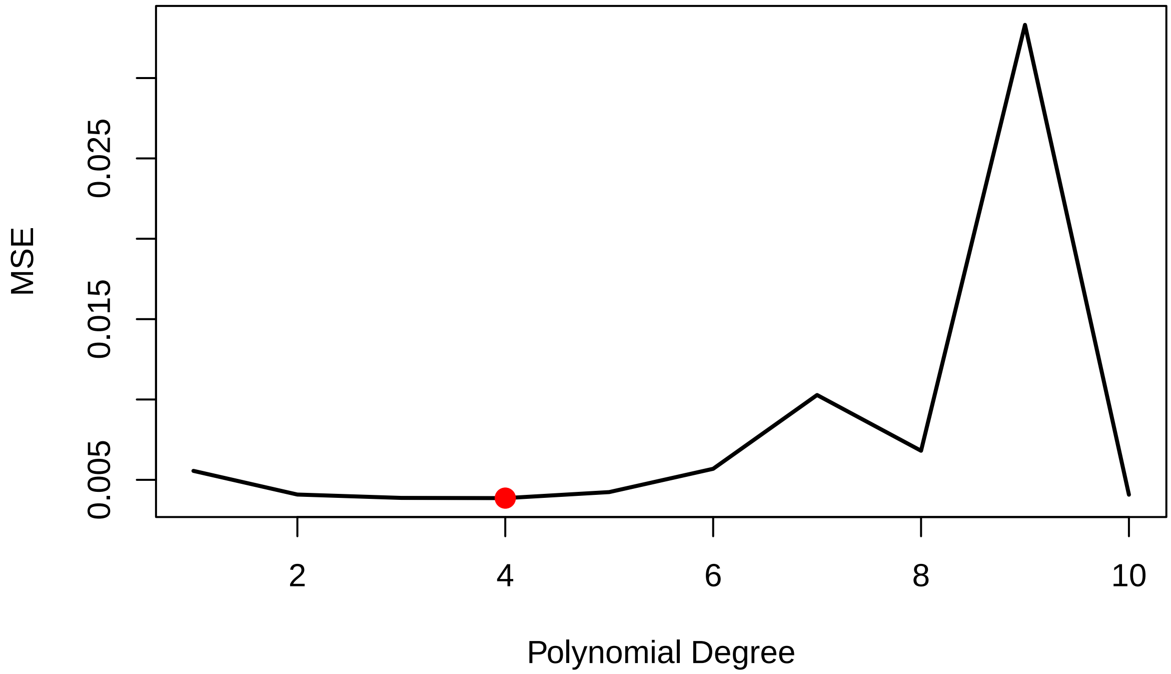
```
set.seed(1)
rss <- rep(NA, 10)
for (i in 1:10){
  fit <- lm(nox ~ poly(dis, i), data = Boston)
  rss[i] <- sum(fit$residuals^2)
}

plot(1:10, rss, xlab = "Polynomial Degree", ylab = "RSS", type = "l", lwd = 2)
points(which.min(rss), rss[which.min(rss)], col='red',pch=20,cex=2)
```



The minimum RSS is for a polynomial degree of 10.

(c)

```
err <- rep(NA, 10)
for (i in 1:10) {
  fit <- glm(nox ~ poly(dis, i), data = Boston)
  err[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(1:10, err, xlab = "Polynomial Degree", ylab = "MSE", type = "l", lwd = 2)
points(which.min(err), err[which.min(err)], col='red',pch=20,cex=2)
```



MSE is the smallest for the polynomial with degree 3.

(d)

```
summary(Boston$dis)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   1.130   2.100   3.207   3.795   5.188  12.127
```

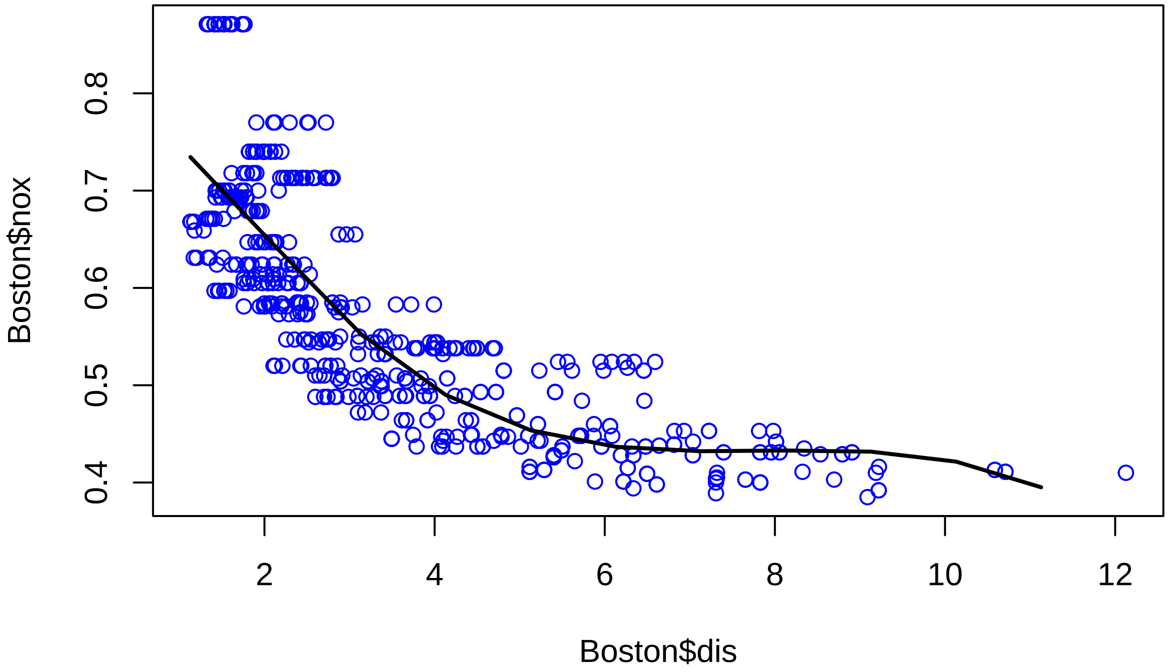
```
fit7.9d <- lm(nox ~ bs(dis, df = 4), data = Boston)
summary(fit7.9d)
```

```
## 
## Call:
## lm(formula = nox ~ bs(dis, df = 4), data = Boston)
## 
## 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
```

```
attr(bs(Boston$dis, df = 4), "knots")
```

```
##     50%
## 3.20745
```

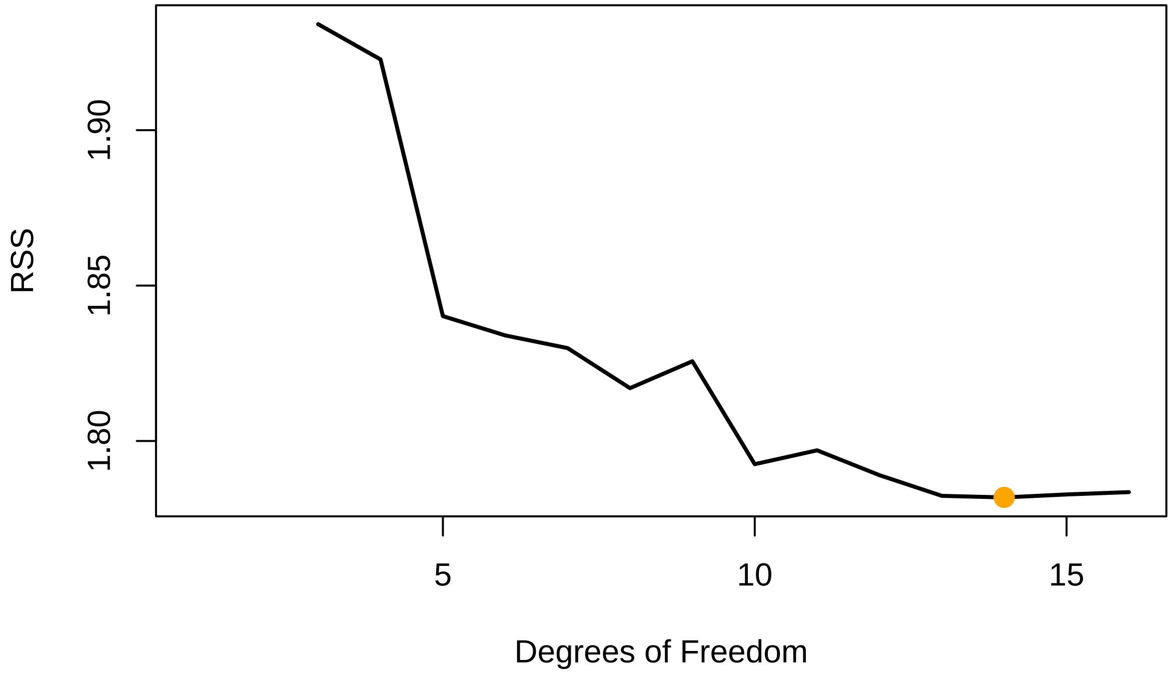
```
x <- seq(min(Boston$dis), max(Boston$dis))
y <- predict(fit7.9d, data.frame(dis = x))
plot(Boston$dis, Boston$nox, col = "blue")
lines(x, y, lwd = 2)
```



Since we chose the degrees of freedom, R chose the knot at 3.207. This corresponds to the 50th percentile of the weighted mean of distances.

(e)

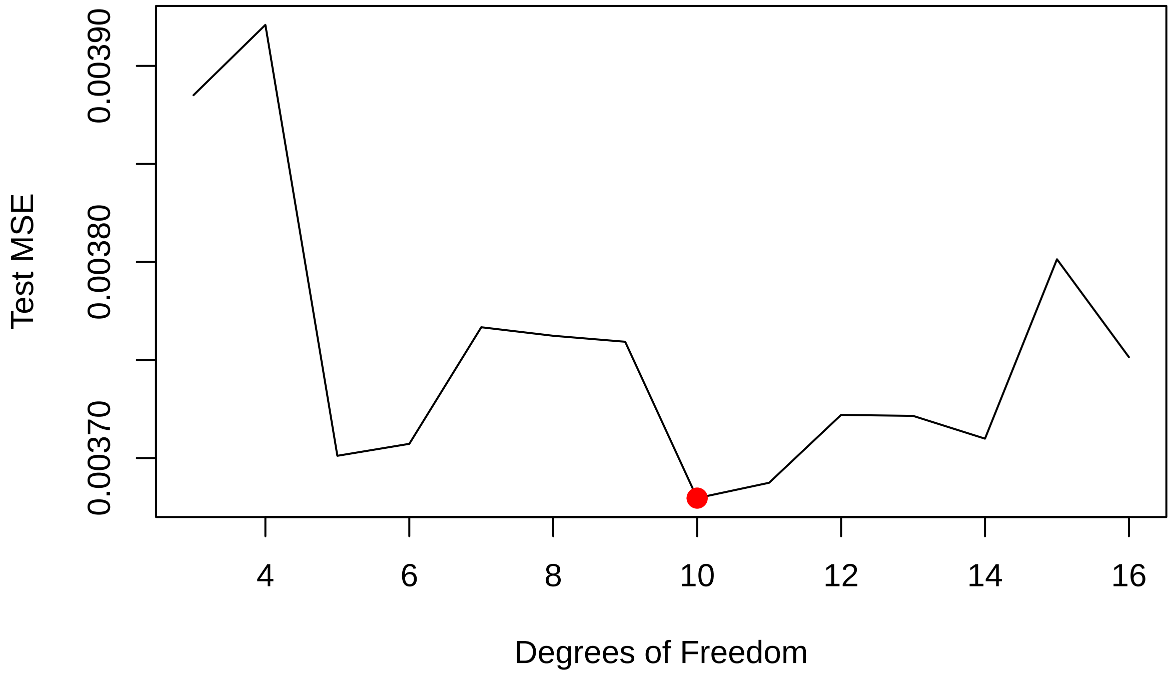
```
rss_df <- c()
for (i in 3:16) {
  fit <- lm(nox ~ bs(dis, df = i), data = Boston)
  pred <- predict(fit, data.frame(dis = x))
  rss_df[i] <- sum(fit$residuals^2)
}

plot(1:16, rss_df, xlab = "Degrees of Freedom", ylab = "RSS", type = "l", lwd = 2)
points(which.min(rss_df), rss_df[which.min(rss_df)], col='orange',pch=20,cex=2)
```



The smallest RSS is for 14 degrees of freedom

(f)

```
set.seed(9)
cv <- rep(NA, 16)
for (i in 3:16) {
  fit <- glm(nox ~ bs(dis, df = i), data = Boston)
  cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}

plot(3:16, cv[3:16], xlab = "Degrees of Freedom", ylab = "Test MSE", type = "l")
points(which.min(cv), cv[which.min(cv)], col = "red", pch = 20, cex = 2)
```



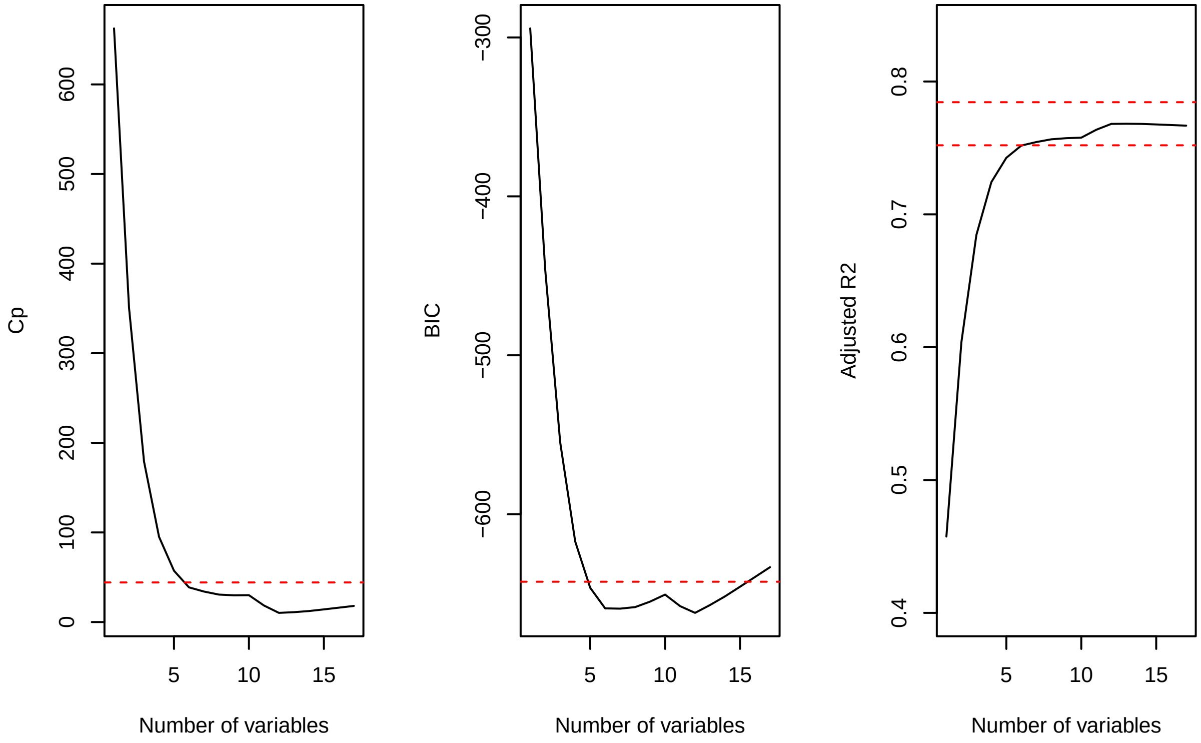
CV shows the smallest test MSE for 10 degrees of freedom.

10. This question relates to the College data set.

(a) Split the data into a training set and a test set. Using out-of-state tuition as the response and the other variables as the predictors, perform forward stepwise selection on the training set in order to identify a satisfactory model that uses just a subset of the predictors.

```
data("College")

set.seed(1)
train_id <- sample(1:nrow(College), 500)
train <- College[train_id,]
test <- College[-train_id,]

fit_fwd <- regsubsets(Outstate ~ ., train, nvmax = ncol(College)-1, method = "forward")
fwd_summary <- summary(fit_fwd)

par(mfrow = c(1, 3))

plot(fwd_summary$cp, xlab = "Number of variables", ylab = "Cp", type = "l")
min.cp <- min(fwd_summary$cp)
std.cp <- sd(fwd_summary$cp)
abline(h = min.cp + 0.2 * std.cp, col = "red", lty = 2)
abline(h = min.cp - 0.2 * std.cp, col = "red", lty = 2)
```

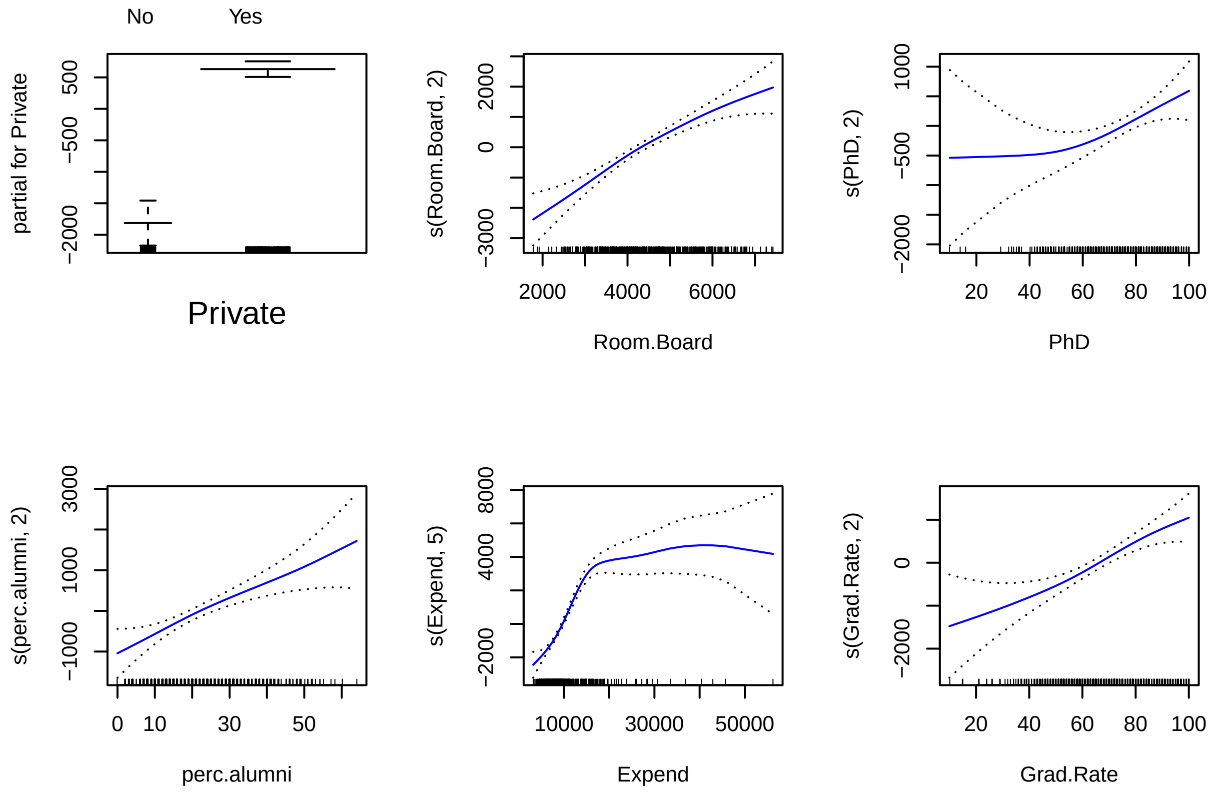
```
plot(fwd_summary$bic, xlab = "Number of variables", ylab = "BIC", type='l')
min.bic <- min(fwd_summary$bic)
std.bic <- sd(fwd_summary$bic)
abline(h = min.bic + 0.2 * std.bic, col = "red", lty = 2)
abline(h = min.bic - 0.2 * std.bic, col = "red", lty = 2)

plot(fwd_summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R2", type = "l", ylim = c(0.4, 0
max.adjr2 <- max(fwd_summary$adjr2)
std.adjr2 <- sd(fwd_summary$adjr2)
abline(h = max.adjr2 + 0.2 * std.adjr2, col = "red", lty = 2)
abline(h = max.adjr2 - 0.2 * std.adjr2, col = "red", lty = 2)
```



The model metrics do not seem to improve much after 6 predictors.

(b)

```
fit.gam <- gam(Outstate ~ Private + s(Room.Board,2) + s(PhD,2) + s(perc.alumni,2) + s(Expend,5) + s(Grad
par(mfrow = c(2,3))
plot(fit.gam, se = TRUE, col = "blue")
```



(c)

```
preds <- predict(fit.gam, test)
err <- mean((test$Outstate - preds)^2)

tss <- mean((test$Outstate - mean(test$Outstate))^2)
rss <- 1 - err / tss
rss
```

```
## [1] 0.7623157
```

The R squared for the GAM model with 6 predictors is 0.7623 or 76.23% of the variation in the model is explained by the predictors chosen. This seems to be quite a good model.

(d) For which variables, if any, is there evidence of a non-linear relationship with the response?

```
summary(fit.gam)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board, 2) + s(PhD,
##     2) + s(perc.alumni, 2) + s(Expend, 5) + s(Grad.Rate, 2),
##     data = train)
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -7650.93 -1148.20    41.73  1320.85  7690.91
##
## (Dispersion Parameter for gaussian family taken to be 3610393)
##
##     Null Deviance: 8686699532 on 499 degrees of freedom
## Residual Deviance: 1751040814 on 485.0001 degrees of freedom
## AIC: 8985.372
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##                    Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private             1 2243338552 2243338552 621.356 < 2.2e-16 ***
## s(Room.Board, 2)    1 1830060351 1830060351 506.887 < 2.2e-16 ***
## s(PhD, 2)           1  529672861  529672861 146.708 < 2.2e-16 ***
## s(perc.alumni, 2)   1  406598388  406598388 112.619 < 2.2e-16 ***
## s(Expend, 5)        1  697931974  697931974 193.312 < 2.2e-16 ***
## s(Grad.Rate, 2)     1   93776215   93776215  25.974 4.968e-07 ***
## Residuals         485 1751040814    3610393
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##                   Npar Df  Npar F   Pr(F)
## (Intercept)
## Private
## s(Room.Board, 2)        1  3.5062 0.06174 .
## s(PhD, 2)               1  1.9648 0.16164
## s(perc.alumni, 2)       1  0.5880 0.44353
## s(Expend, 5)            4 23.8305 < 2e-16 ***
## s(Grad.Rate, 2)         1  1.5954 0.20716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Expend has a strong non-linear relationship with the response variable.

11.

(a)

```
set.seed(1)
y <- rnorm(100)
x1 <- rnorm(100)
x2 <- rnorm(100)
```

(b)

```
b_h1 <- 1
```

(c)

```r
a <- y - b_h1*x1
b_h2 <- lm(a~x2)$coef[2]
```

(d)

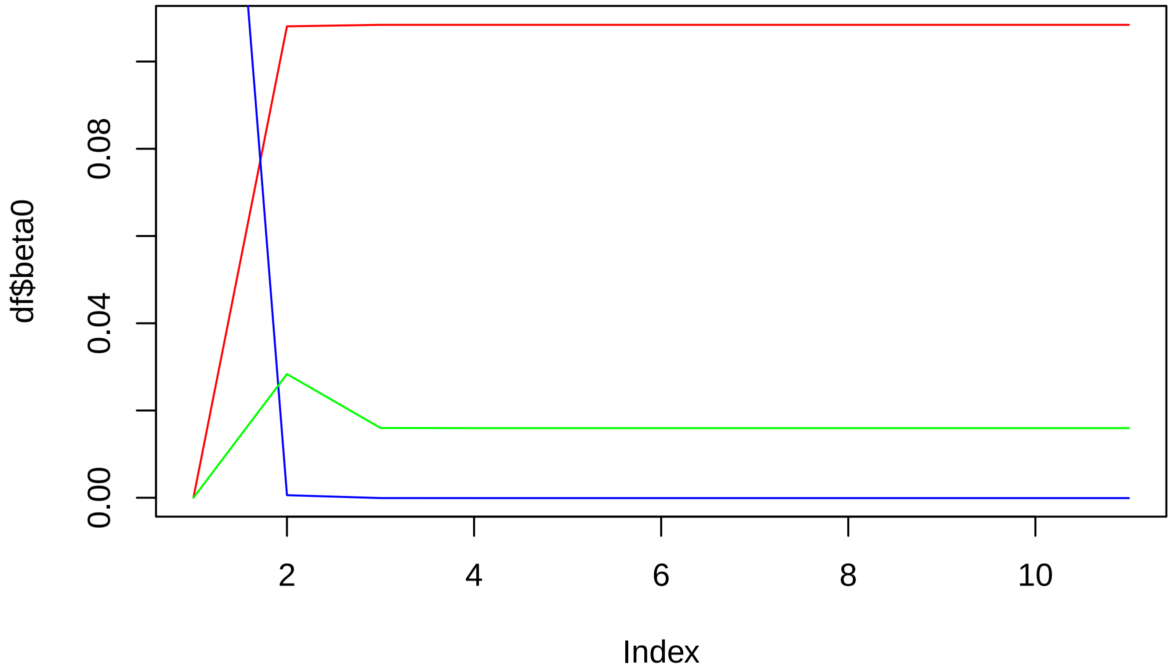
```r
a <- y- b_h2*x2
b_h1 <- lm(a~x1)$coef[2]
```

(e)

```r
iter <- 10
df <- data.frame(0.0, 0.27, 0.0)
names(df) <- c('beta0', 'beta1', 'beta2')
for (i in 1:iter) {
  beta1 <- df[nrow(df), 2]
  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)
  df[nrow(df) + 1,] <- list(beta0, beta1, beta2)
}
```

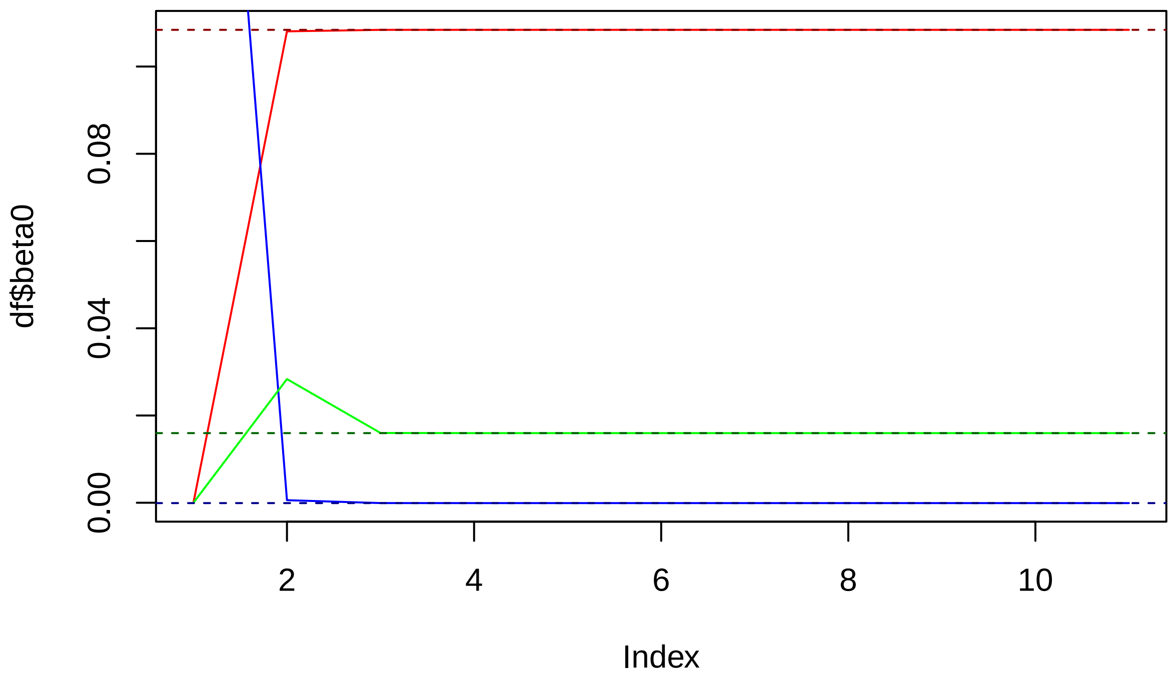
```
## (Intercept)
##   0.1080682
##          x1
## 0.000584017
##         x2
## 0.02835083
## (Intercept)
##     0.10841
##            x1
## -7.708576e-05
##         x2
## 0.01599065
## (Intercept)
##   0.1084108
##          x1
## -7.8708e-05
##         x2
## 0.01596032
## (Intercept)
##   0.1084108
##            x1
## -7.871198e-05
##         x2
## 0.01596025
## (Intercept)
##   0.1084108
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025 
## (Intercept) 
##   0.1084108 
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025 
## (Intercept) 
##   0.1084108 
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025 
## (Intercept) 
##   0.1084108 
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025 
## (Intercept) 
##   0.1084108 
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025 
## (Intercept) 
##   0.1084108 
##            x1 
## -7.871199e-05 
##         x2 
## 0.01596025
```

```r
plot(df$beta0, col = 'red', type = 'l')
lines(df$beta1, col = 'blue')
lines(df$beta2, col = 'green')
```



(f)

```
plot(df$beta0, col = 'red', type = 'l')
lines(df$beta1, col = 'blue')
lines(df$beta2, col = 'green')
res <- coef(lm(y ~ x1 + x2))
abline(h = res[1], col = 'darkred', lty = 2)
abline(h = res[2], col = 'darkblue', lty = 2)
abline(h = res[3], col = 'darkgreen', lty = 2)
```


df$beta0
0.00
0.04
0.08
2
4
6
8
10
Index