Smoothing homework

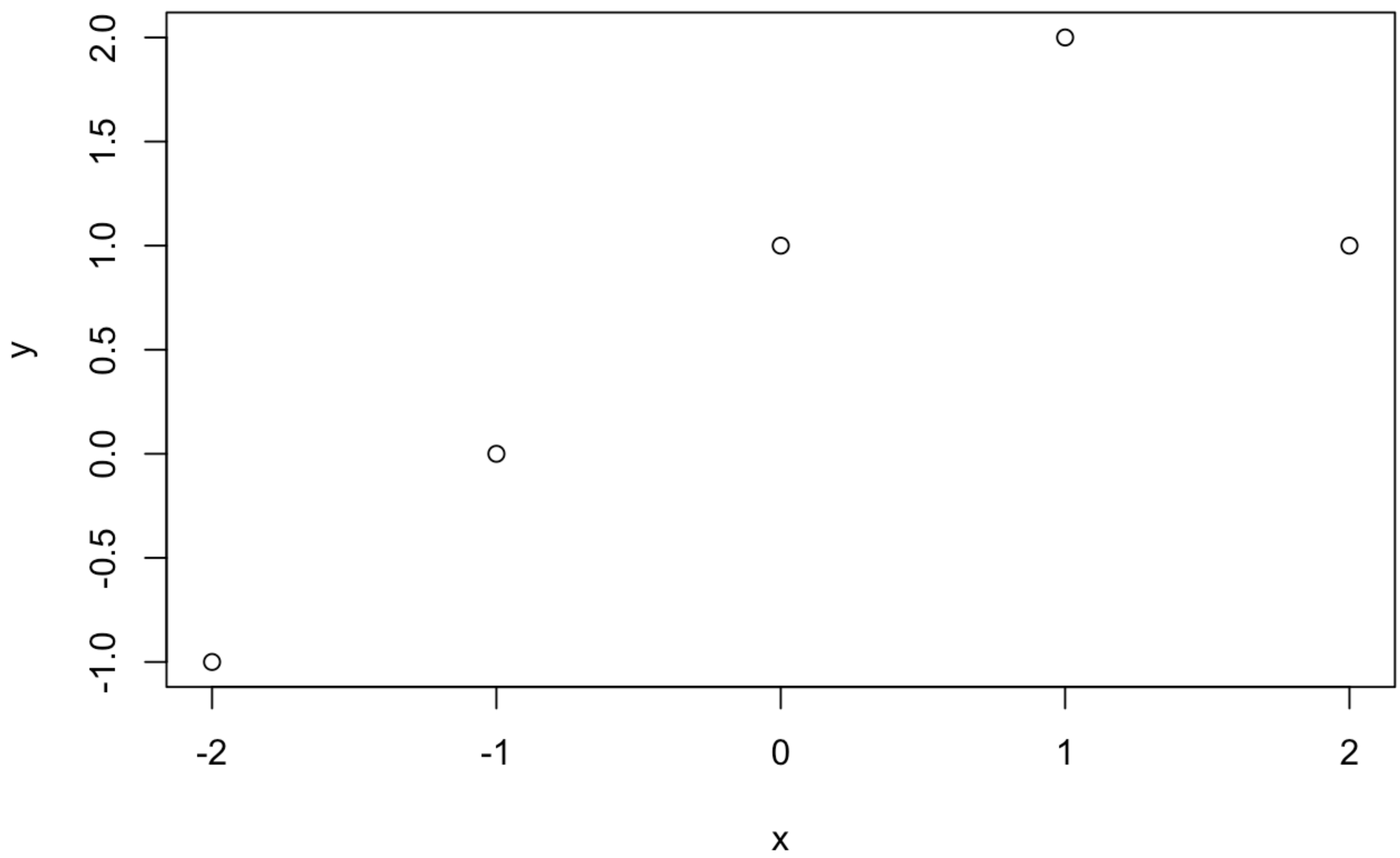
Gunangyan

2/28/2019

```
library(ggplot2)
library(boot)
library(splines)
```

Q3 Suppose we fit a curve with basis functions b1(X)=X, b2(X)=(X − 1)2I(X ≥ 1). (Note that I(X ≥ 1) equals 1 for X ≥ 1and0 otherwise.) We fit the linear regression model

```
x = -2:2
y = 1+x+-2*(x-1)^2*I(x>1)
plot(x,y)
```



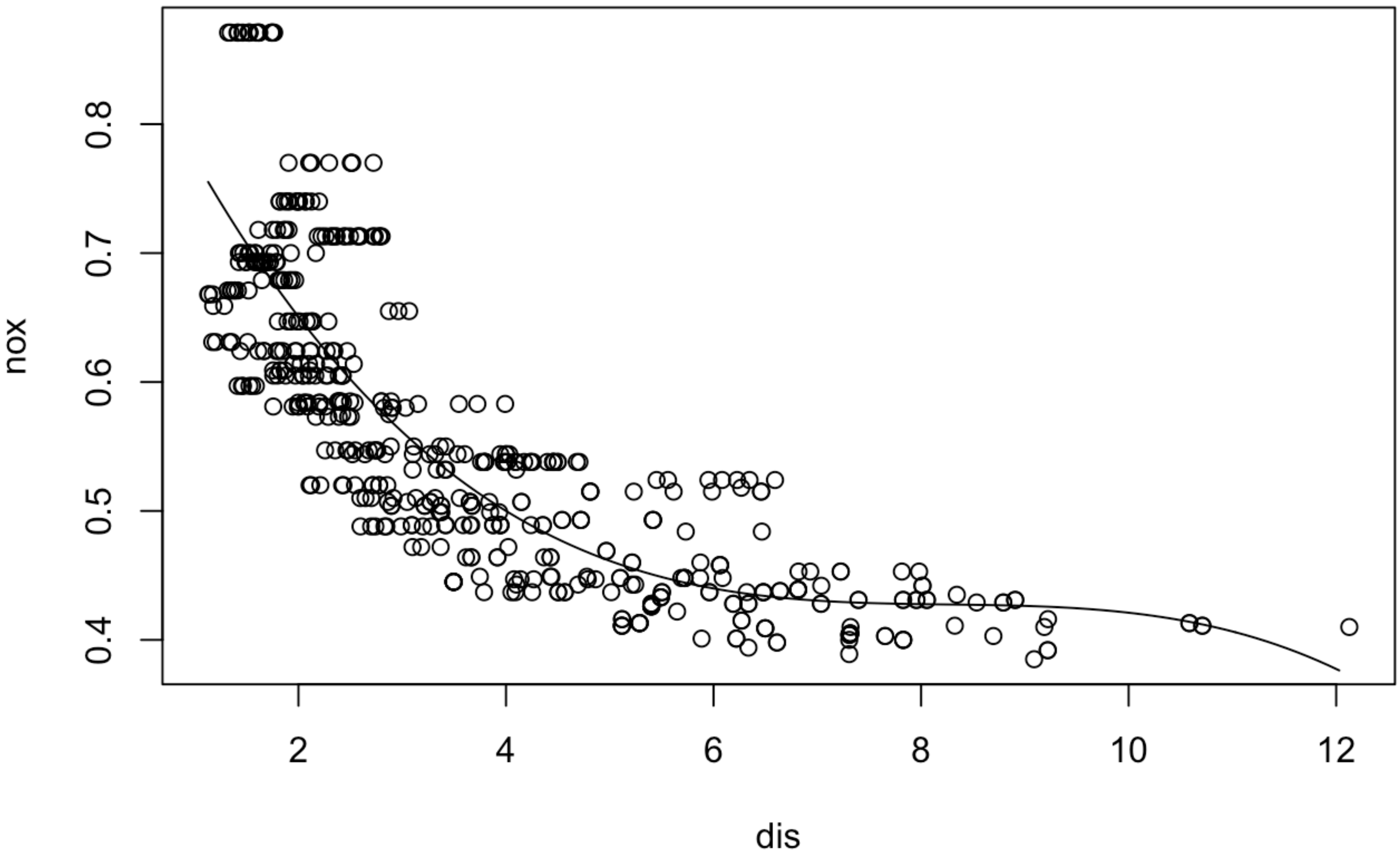
The curve is linear between −2 and 1: y=1+x and quadratic between 1 and 2: y=1+x−2(x−1)2

Q9

```
library(MASS)
set.seed(1)
fit1 <- lm(nox~poly(dis,3),data = Boston)
summary(fit1)
```

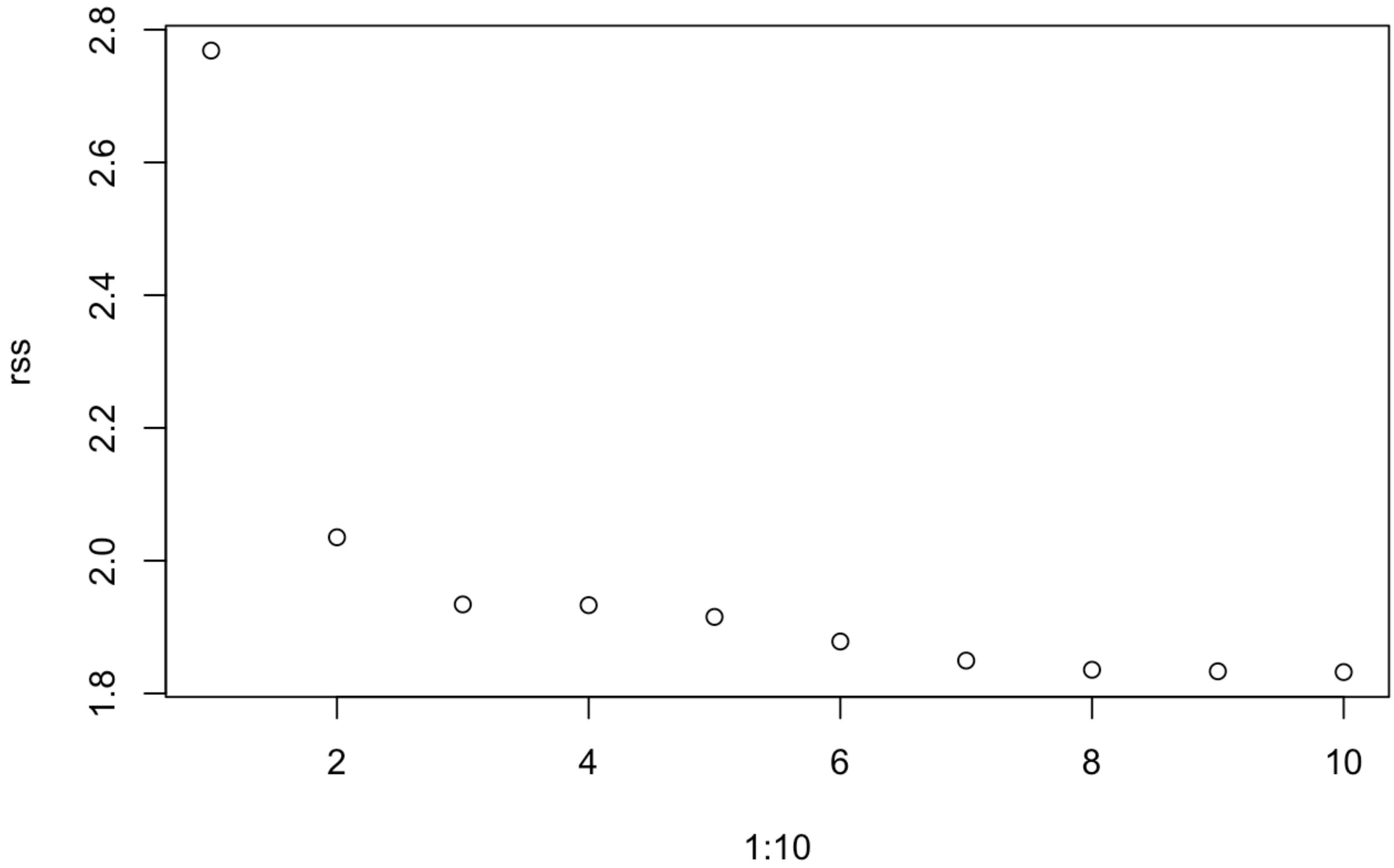
```
##
## Call:
## lm(formula = nox ~ poly(dis, 3), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.121130 -0.040619 -0.009738  0.023385  0.194904
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.554695   0.002759 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
```

```
dislims<-range(Boston$dis)
dis.grid <- seq(from = dislims[1], to = dislims[2], by = 0.1)
preds <- predict(fit1, list(dis = dis.grid))
plot(nox~dis,data=Boston)
lines(x=dis.grid,y=preds)
```



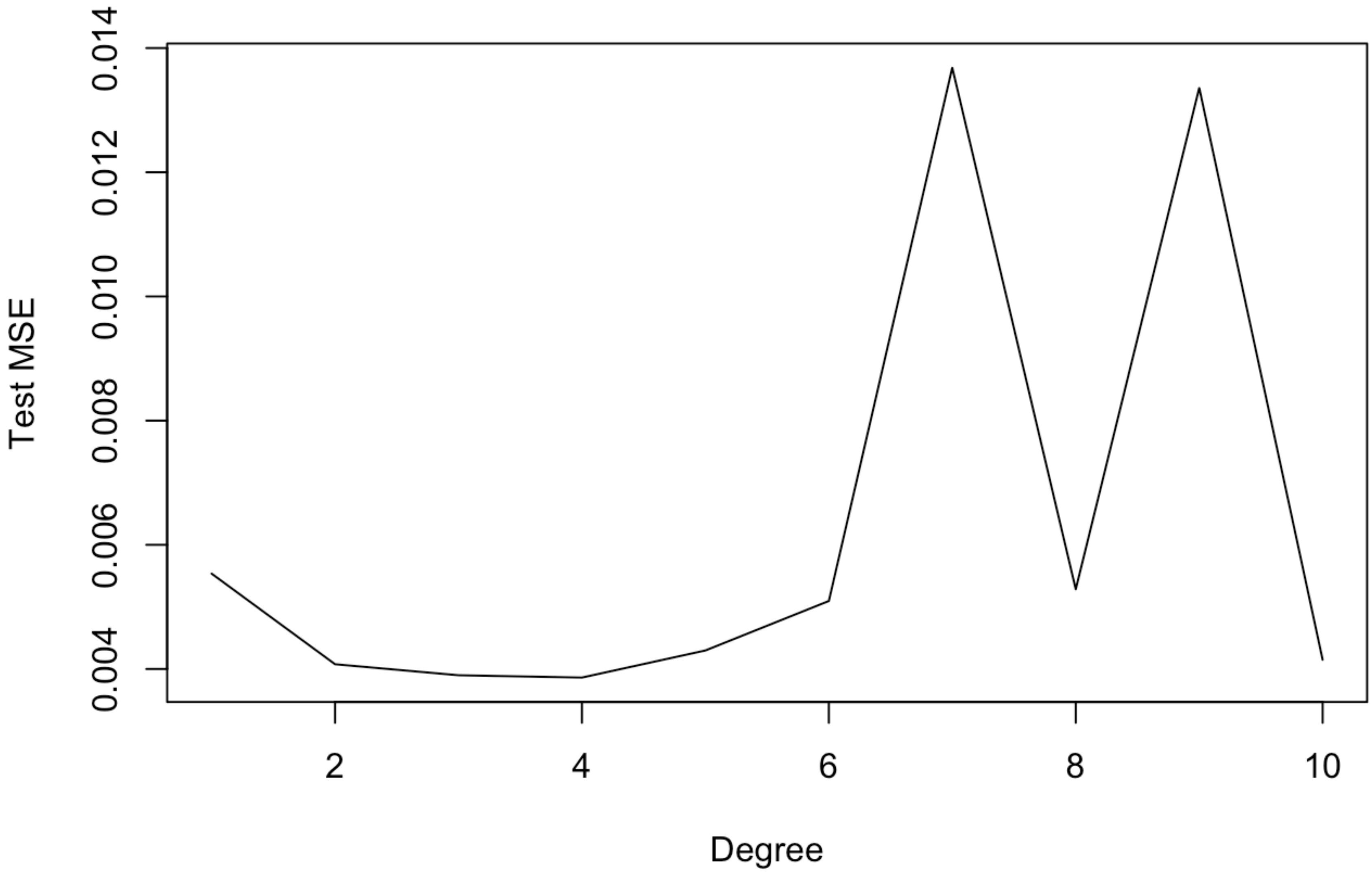

```
rss<-rep(NA,10)
for (i in 1:10) {
  fit<-lm(nox~poly(dis,i),data = Boston)
  rss[i]<-sum(fit$residuals^2)

}
plot(x=1:10,y=rss)
```



The residuals sum of squares decreases.

```
deltas <- rep(NA, 10)
for (i in 1:10) {
    fit <- glm(nox ~ poly(dis, i), data = Boston)
    deltas[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(1:10, deltas, xlab = "Degree", ylab = "Test MSE", type = "l")
```



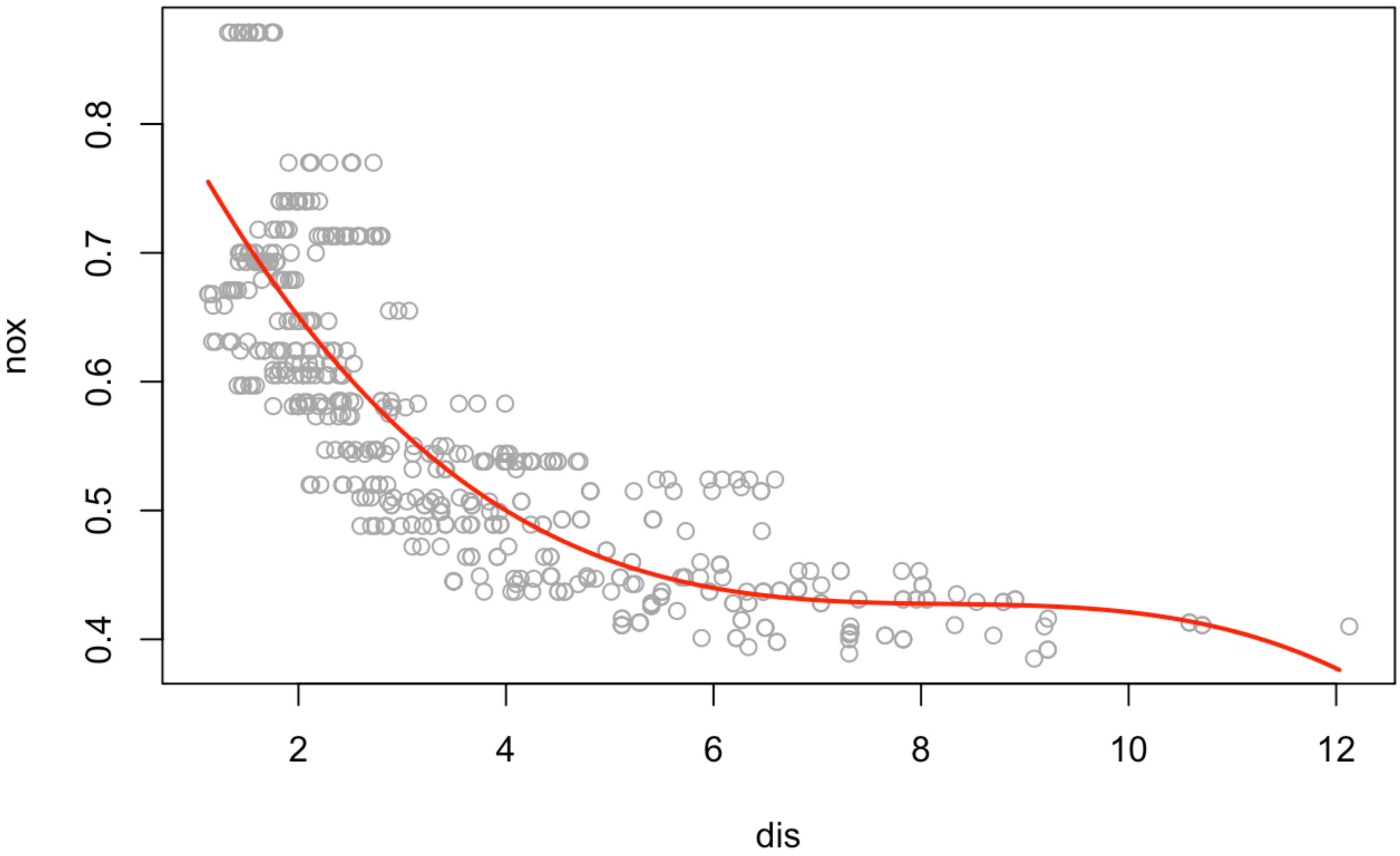
###The degree 4 minimizes MSE.

D

```
fit <- lm(nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)
summary(fit)
```

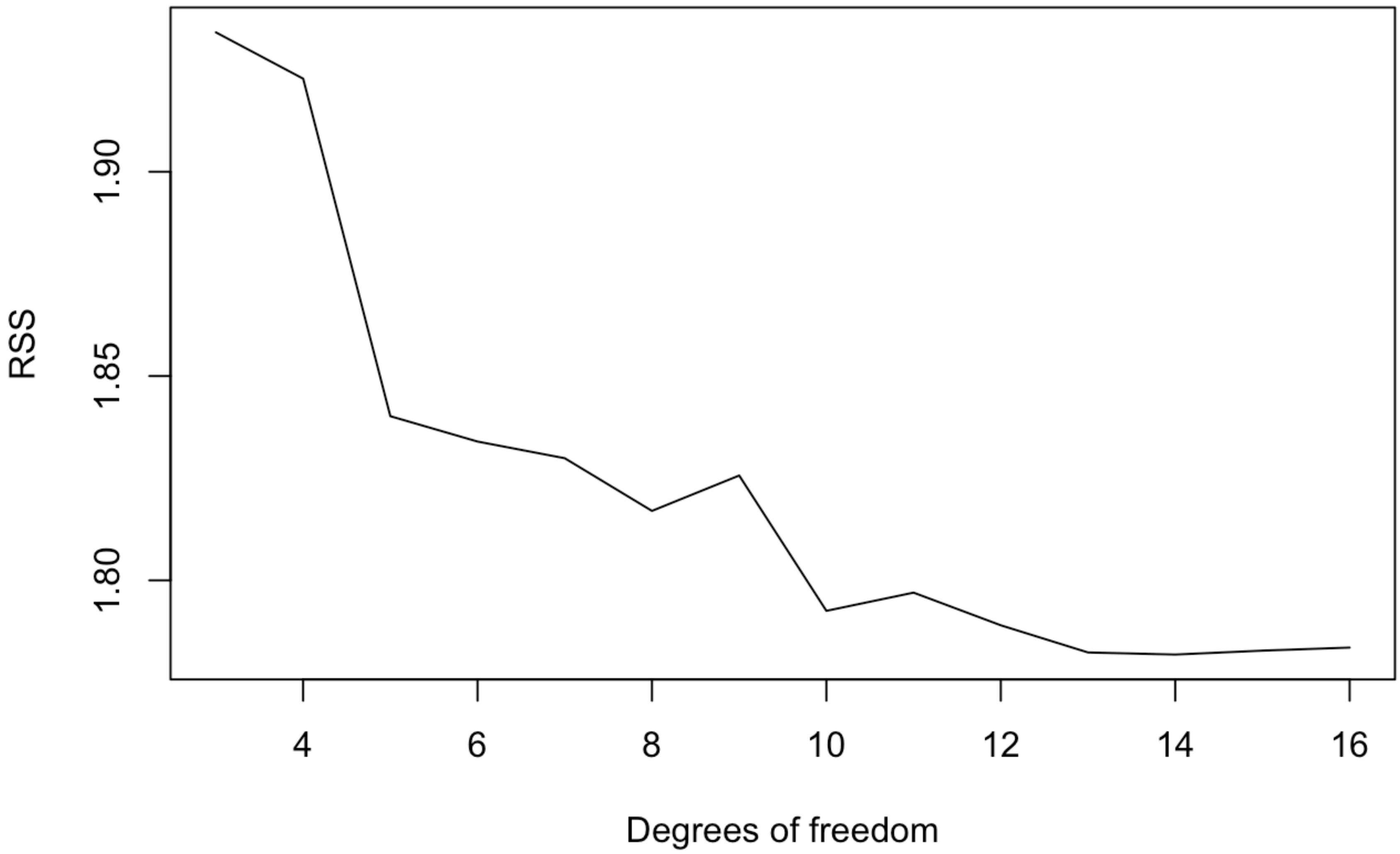
```
##
## Call:
## lm(formula = nox ~ bs(dis, knots = c(4, 7, 11)), data = Boston)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.124567 -0.040355 -0.008702  0.024740  0.192920
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.73926    0.01331  55.537  < 2e-16 ***
## bs(dis, knots = c(4, 7, 11))1 -0.08861    0.02504  -3.539  0.00044 ***
## bs(dis, knots = c(4, 7, 11))2 -0.31341    0.01680 -18.658  < 2e-16 ***
## bs(dis, knots = c(4, 7, 11))3 -0.26618    0.03147  -8.459 3.00e-16 ***
## bs(dis, knots = c(4, 7, 11))4 -0.39802    0.04647  -8.565  < 2e-16 ***
## bs(dis, knots = c(4, 7, 11))5 -0.25681    0.09001  -2.853  0.00451 **
## bs(dis, knots = c(4, 7, 11))6 -0.32926    0.06327  -5.204 2.85e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06185 on 499 degrees of freedom
## Multiple R-squared:  0.7185, Adjusted R-squared:  0.7151
## F-statistic: 212.3 on 6 and 499 DF,  p-value: < 2.2e-16
```

```
pred <- predict(fit, list(dis = dis.grid))
plot(nox ~ dis, data = Boston, col = "darkgrey")
lines(dis.grid, preds, col = "red", lwd = 2)
```



E

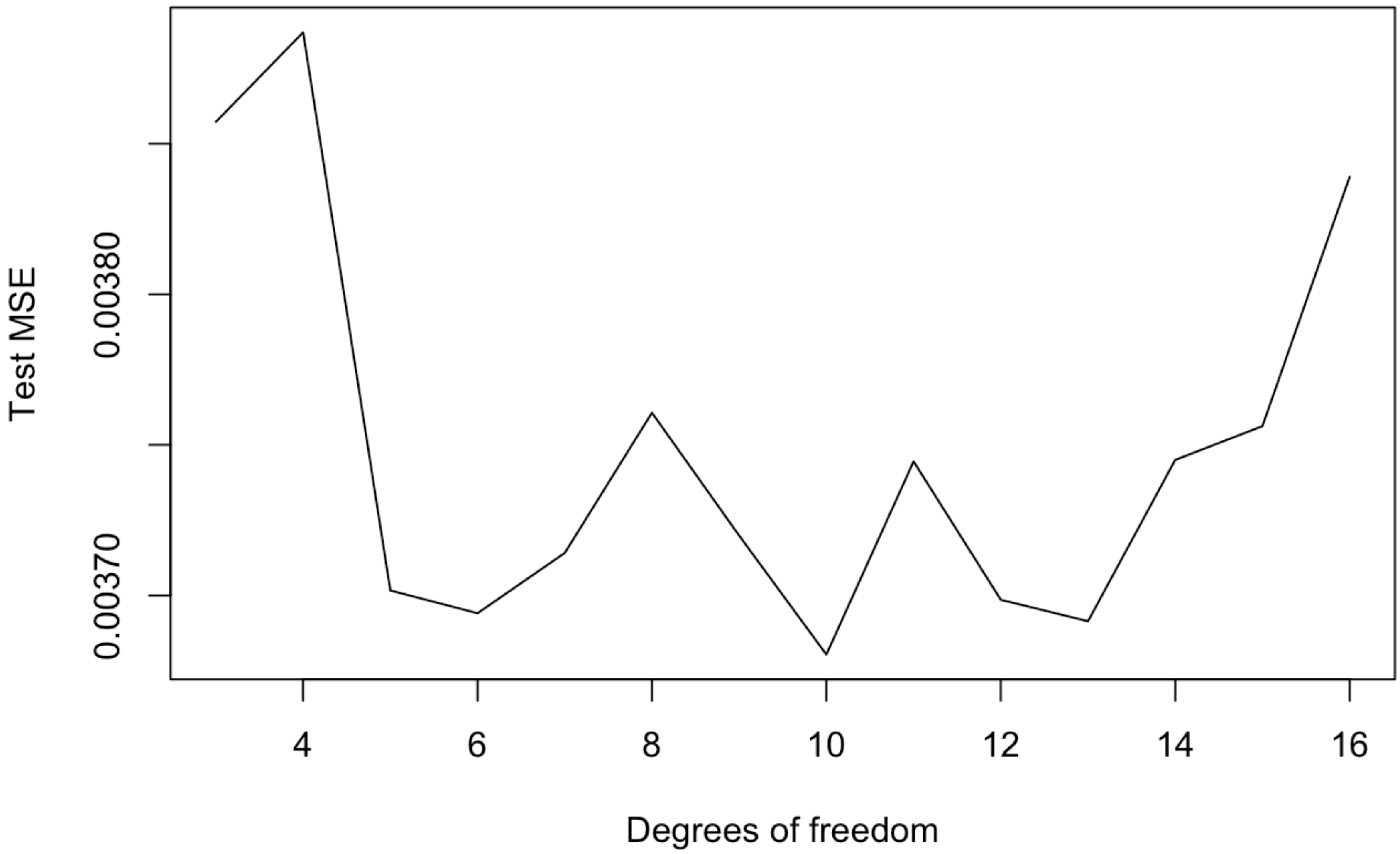
```
rss <- rep(NA, 16)
for (i in 3:16) {
    fit <- lm(nox ~ bs(dis, df = i), data = Boston)
    rss[i] <- sum(fit$residuals^2)
}
plot(3:16, rss[-c(1, 2)], xlab = "Degrees of freedom", ylab = "RSS", type = "l")
```



###It will decrease when freedom reaches to 14 and the increase slightly.

f

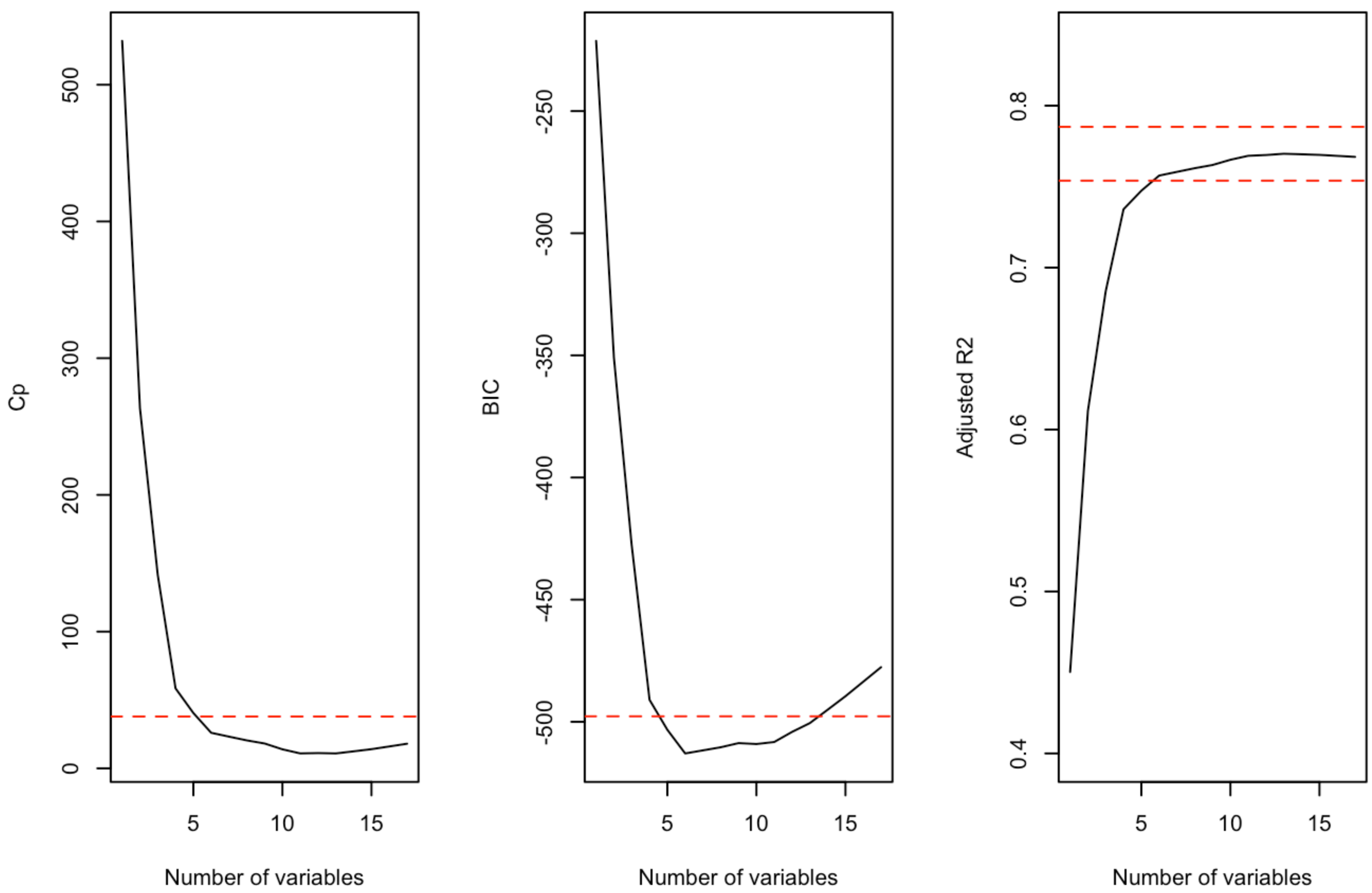
```
cv <- rep(NA, 16)
for (i in 3:16) {
    fit <- glm(nox ~ bs(dis, df = i), data = Boston)
    cv[i] <- cv.glm(Boston, fit, K = 10)$delta[1]
}
plot(3:16, cv[-c(1, 2)], xlab = "Degrees of freedom", ylab = "Test MSE", type = "l")
```



Q10

A

```r
library(leaps)
#set.seed(1)
#save(College_test,file = "College_test.RData")
#save(College_train,file = "College_train.Rdata")
#train <- sample(length(Outstate), length(Outstate) / 2)
#test <- -train
load("College_train.RData")
load("College_test.RData")
fit <- regsubsets(Outstate ~ ., data = College_train, nvmax = 17, method = "forward")
fit.summary <- summary(fit)
par(mfrow = c(1, 3))
plot(fit.summary$cp, xlab = "Number of variables", ylab = "Cp", type = "l")
min.cp <- min(fit.summary$cp)
std.cp <- sd(fit.summary$cp)
abline(h = min.cp + 0.2 * std.cp, col = "red", lty = 2)
abline(h = min.cp - 0.2 * std.cp, col = "red", lty = 2)
plot(fit.summary$bic, xlab = "Number of variables", ylab = "BIC", type='l')
min.bic <- min(fit.summary$bic)
std.bic <- sd(fit.summary$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(fit.summary$adjr2, xlab = "Number of variables", ylab = "Adjusted R2", type = "l
", ylim = c(0.4, 0.84))
max.adjr2 <- max(fit.summary$adjr2)
std.adjr2 <- sd(fit.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)
```




```
fit <- regsubsets(Outstate ~ ., data = College_test, method = "forward")
coeffs <- coef(fit, id = 6)
names(coeffs)
```

```
## [1] "(Intercept)" "PrivateYes"   "Room.Board"   "PhD"          "perc.alumni"
## [6] "Expend"      "Grad.Rate"
```

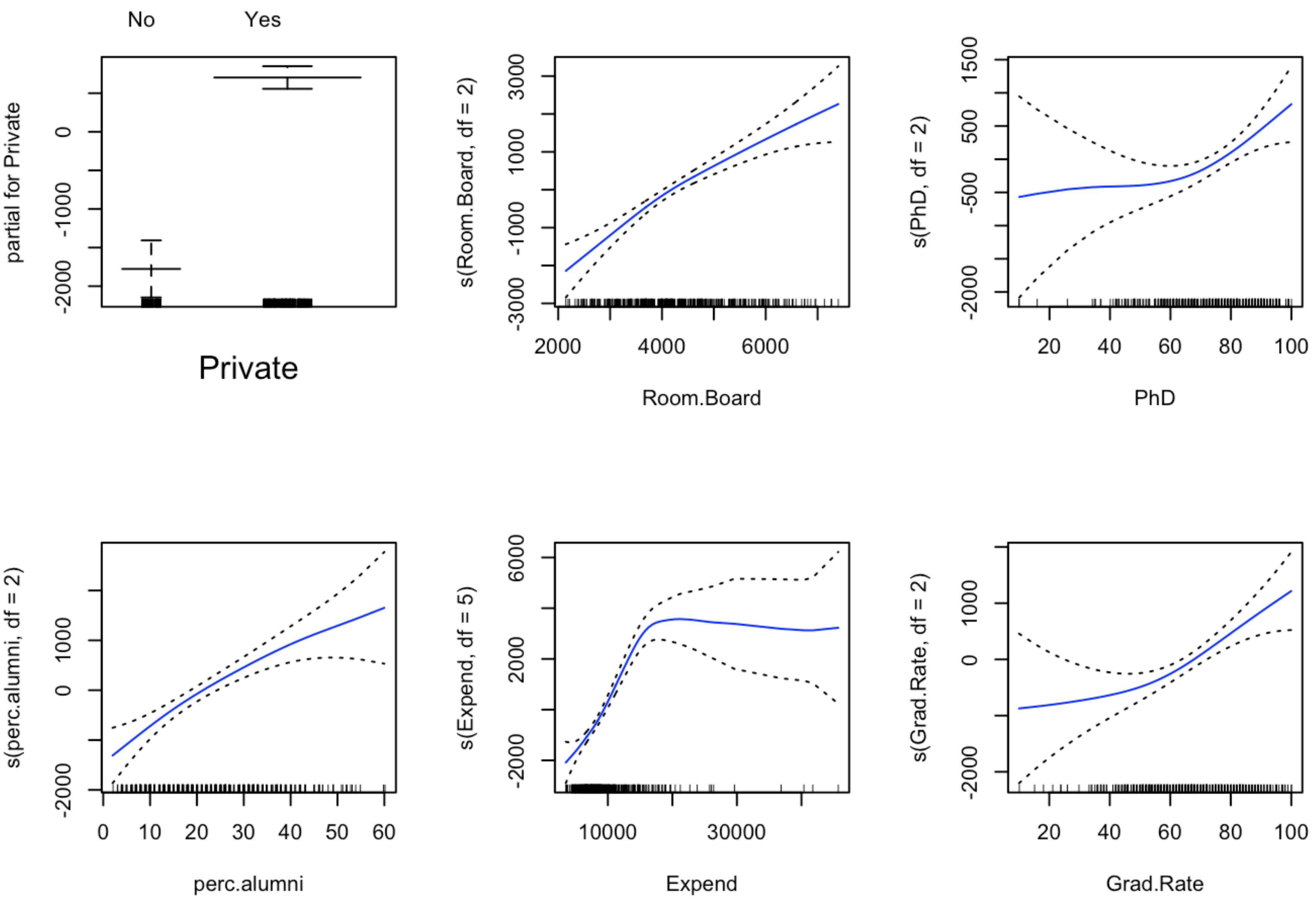
B

```
library(gam)
```

```
## Loading required package: foreach
```

```
## Loaded gam 1.16
```

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



C

```
preds <- predict(fit, College_test)
err <- mean((College_test$Outstate - preds)^2)
err
```

```
## [1] 3745460
```

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

```
## [1] 0.7696916
```

d

```
summary(fit)
```

```
##
## Call: gam(formula = Outstate ~ Private + s(Room.Board, df = 2) + s(PhD,
##     df = 2) + s(perc.alumni, df = 2) + s(Expend, df = 5) + s(Grad.Rate,
##     df = 2), data = College_train)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4977.74 -1184.52    58.33  1220.04  7688.30
##
## (Dispersion Parameter for gaussian family taken to be 3300711)
##
##     Null Deviance: 6221998532 on 387 degrees of freedom
## Residual Deviance: 1231165118 on 373 degrees of freedom
## AIC: 6941.542
##
## Number of Local Scoring Iterations: 2
##
## Anova for Parametric Effects
##                         Df     Sum Sq    Mean Sq F value    Pr(>F)
## Private                  1 1779433688 1779433688 539.106 < 2.2e-16 ***
## s(Room.Board, df = 2)    1 1221825562 1221825562 370.171 < 2.2e-16 ***
## s(PhD, df = 2)           1  382472137  382472137 115.876 < 2.2e-16 ***
## s(perc.alumni, df = 2)   1  328493313  328493313  99.522 < 2.2e-16 ***
## s(Expend, df = 5)        1  416585875  416585875 126.211 < 2.2e-16 ***
## s(Grad.Rate, df = 2)     1   55284580   55284580  16.749 5.232e-05 ***
## Residuals              373 1231165118    3300711
## ---
## 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, df = 2)        1  3.5562   0.06010 .
## s(PhD, df = 2)               1  4.3421   0.03786 *
## s(perc.alumni, df = 2)       1  1.9158   0.16715
## s(Expend, df = 5)            4 16.8636 1.016e-12 ***
## s(Grad.Rate, df = 2)         1  3.7208   0.05450 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

ANOVA shows a strong evidence of non-linear relationship between “Outstate” and “Expend””“, and a moderately strong non-linear relationship (using p-value of 0.05)

between”Outstate" and “Grad.Rate”" or “PhD”.

11

a&b

```
set.seed(1)
y <- rnorm(100)
x1 <- rnorm(100)
x2 <- rnorm(100)
beta1 <- 3
```

c

```
a <- y - beta1 * x1
beta2 <- lm(a ~ x2)$coef[2]
```

d

```
a <- y - beta2 * x2
beta1 <- lm(a ~ x1)$coef[2]
```

e

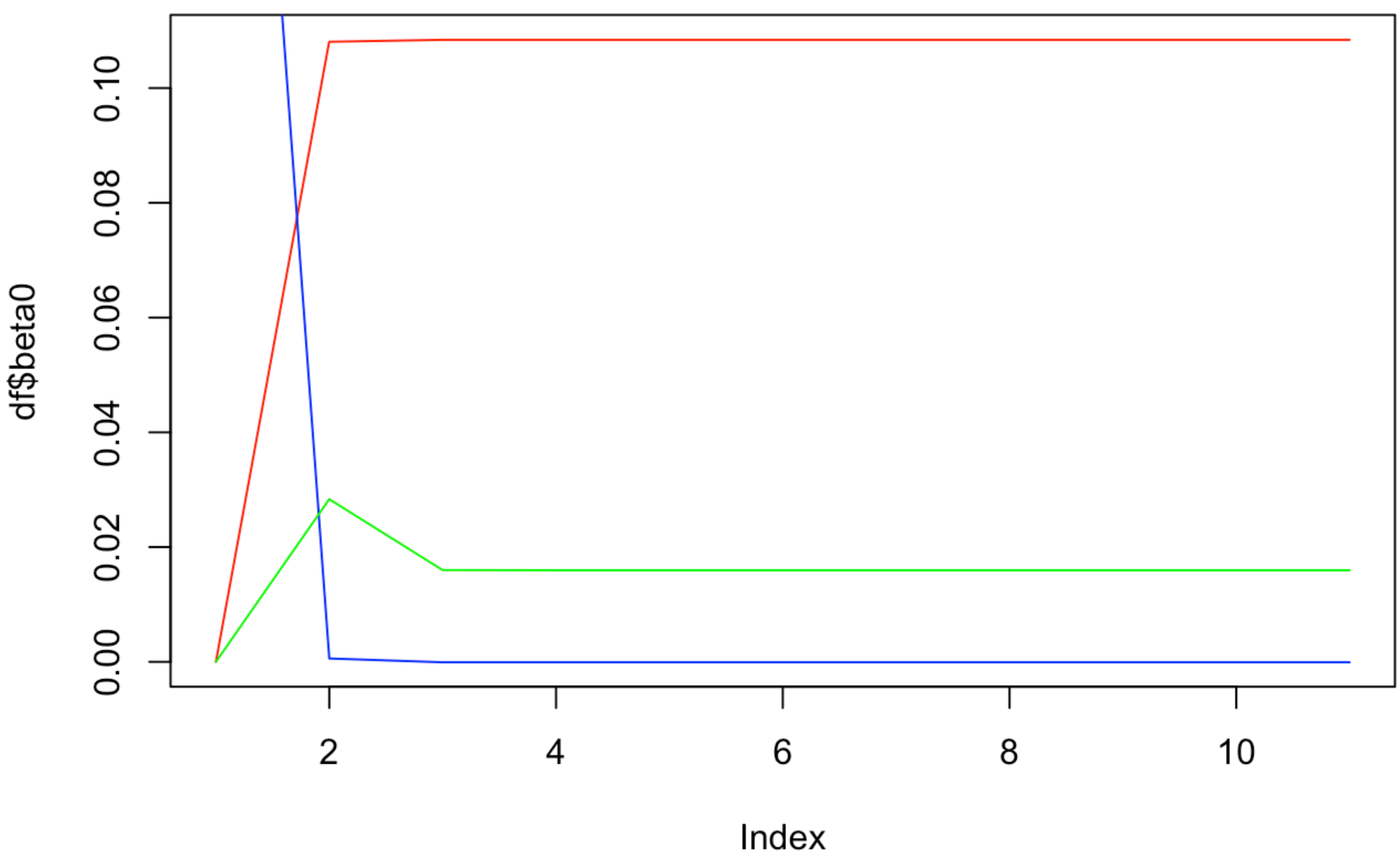
```
iter <- 10
df <- data.frame(0.0, 0.27, 0.0)
names(df) <- c('beta0', 'beta1', 'beta2')
for (i in 1:iter) {
  beta1 <- df[nrow(df), 2]
  a <- y - beta1 * x1
  beta2 <- lm(a ~ x2)$coef[2]
  a <- y - beta2 * x2
  beta1 <- lm(a ~ x1)$coef[2]
  beta0 <- lm(a ~ x1)$coef[1]
  print(beta0)
  print(beta1)
  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
```

```
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)
```