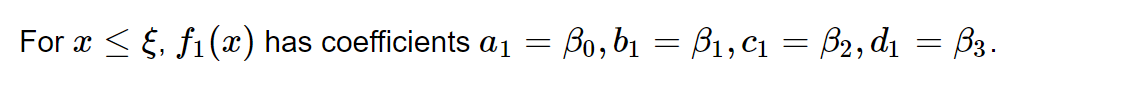
高级统计方法 第8次作业:

序号：27 姓名：王琪瑞 学号：20202241014 班级：软2002

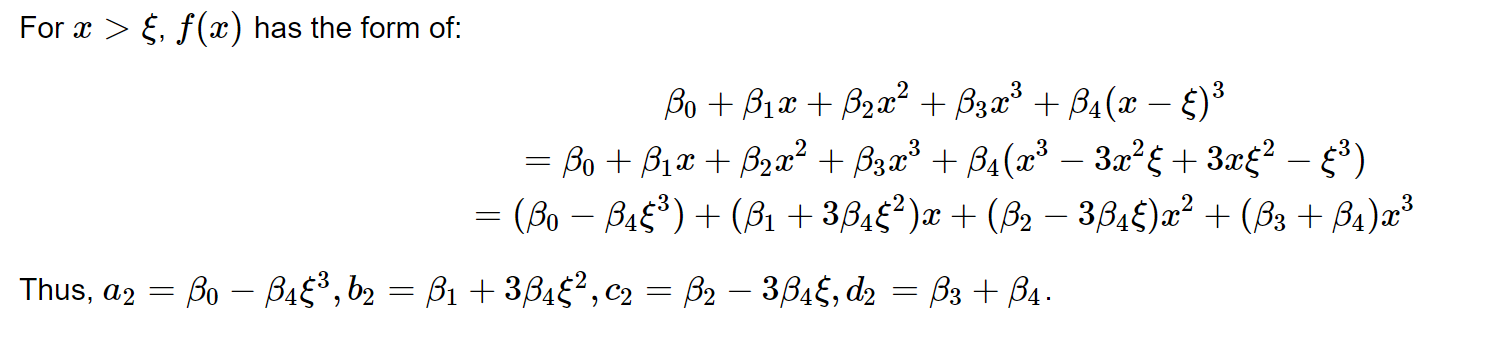
**概念**

1.问题

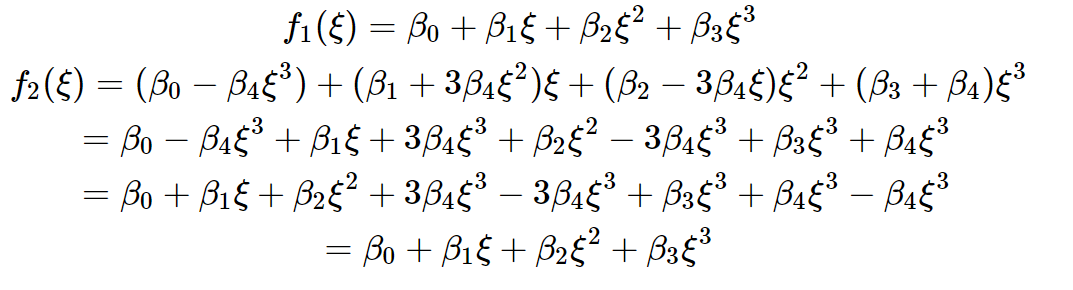
(a)



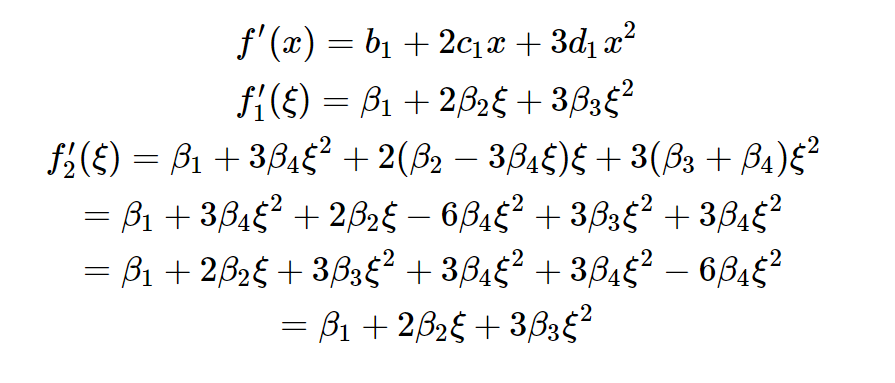
(b)



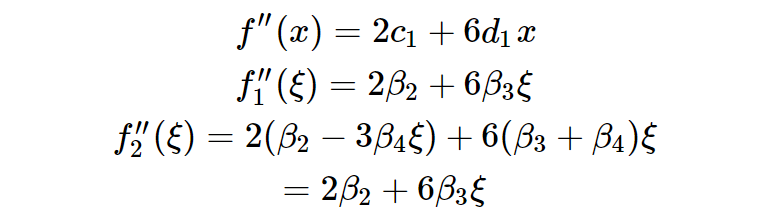
(c)



(d)



(e)



2.问题（略）

（a）问题（略）

λ为无穷，忽略RSS, m=0，此时g应为0，即y=0，使其最小。

（b）问题（略）

λ为无穷，忽略RSS, m=1，此时g应为一条直线，例如y=k，使其最小。

（c）问题（略）

λ为无穷，忽略RSS, m=2，此时g的一阶导数为常数，g为一次函数，例如y=kx+c，使其最小。

（d）问题（略）

λ为无穷，忽略RSS, m=3，此时g的二阶导数为常数，g为二次函数，例如y=kx^2+c，使其最小。

（e）问题（略）

λ为0，忽略第二项, m=3，此时为最小二乘选择g。

3.问题（略）

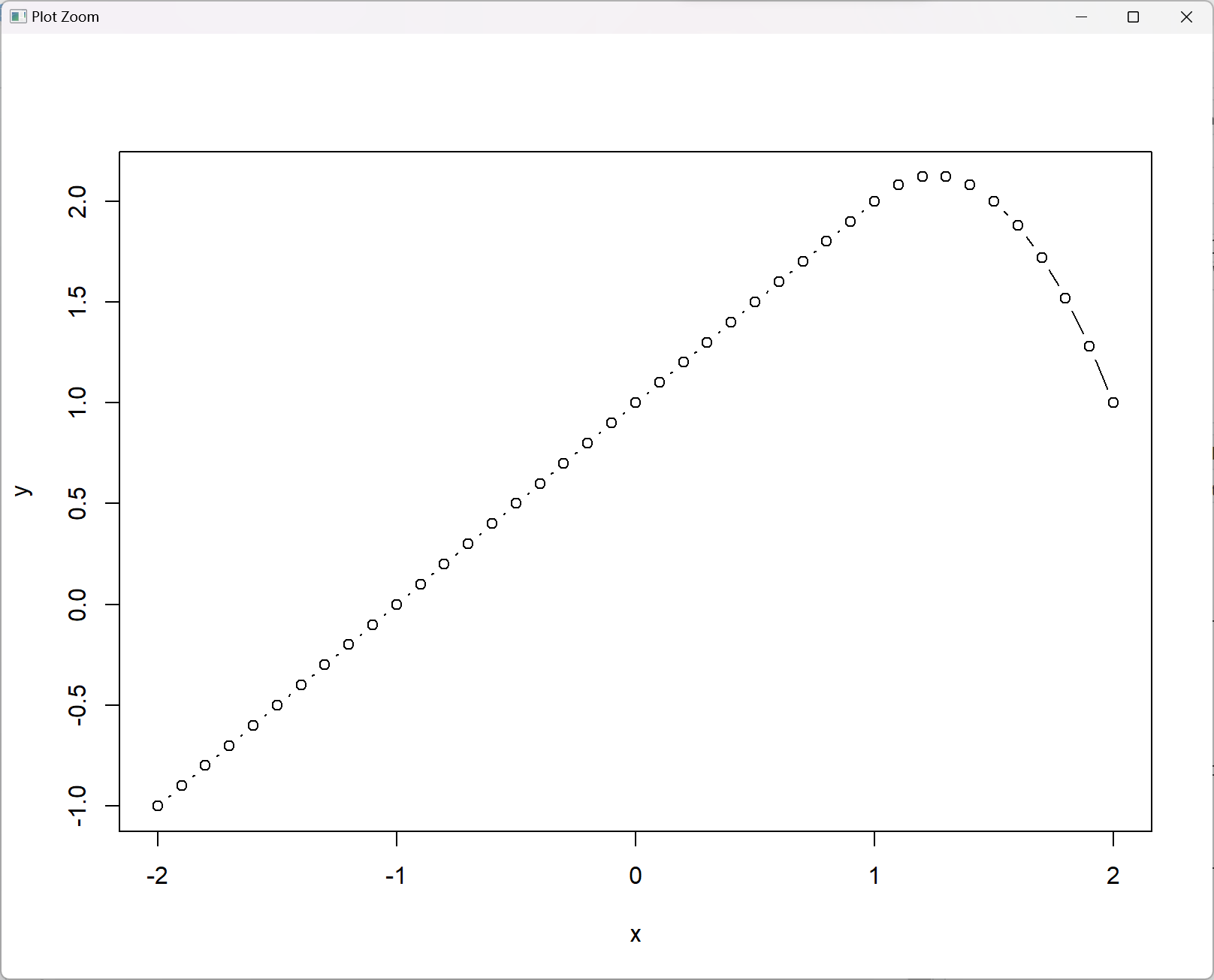
脚本：

x = seq(-2,2,0.1)

y = 1 + x -2 \* (x-1)^2 \* I(x>=1)

plot(x,y,type="b")

截图：



4.问题（略）

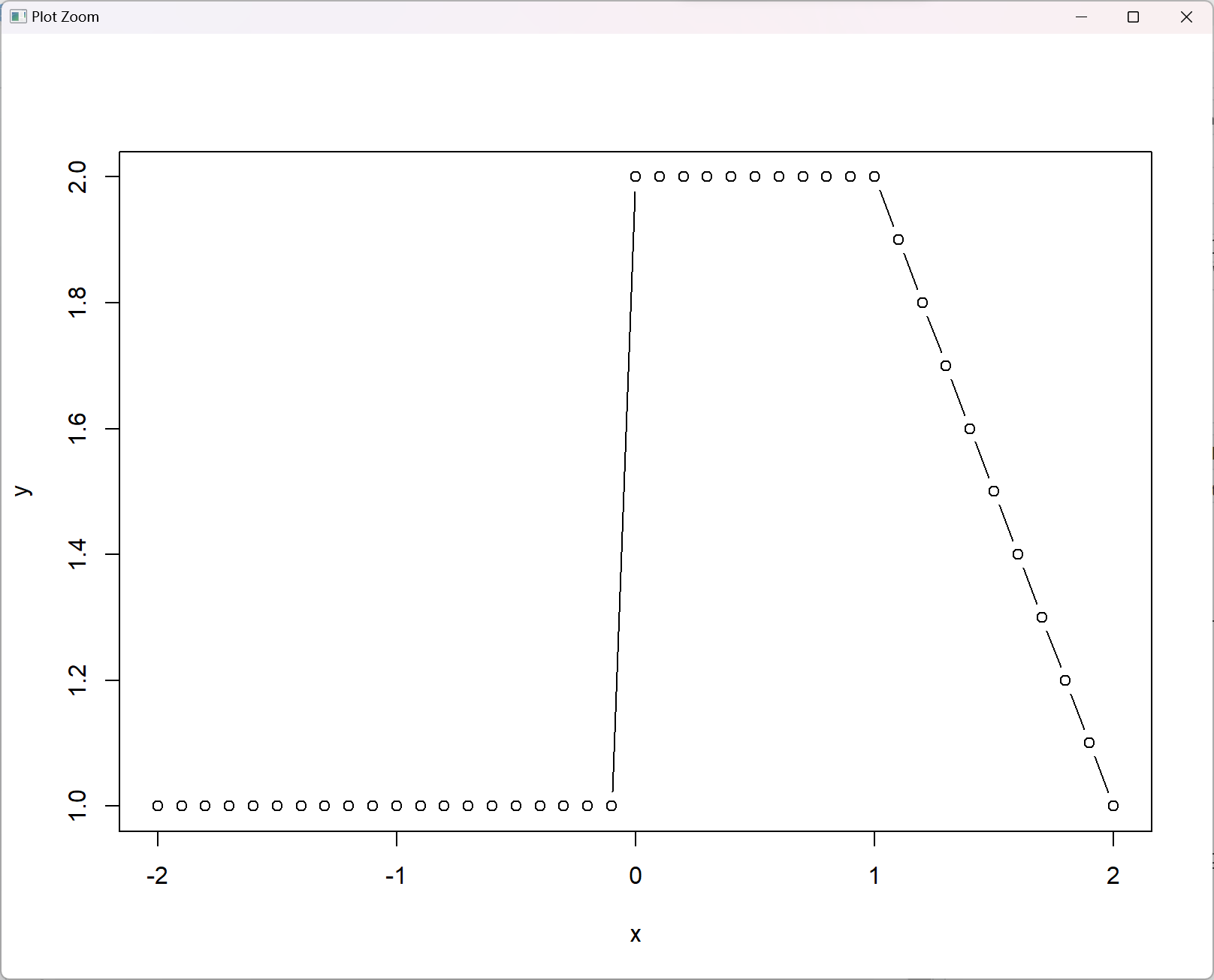
脚本：

x = seq(-2,2,0.1)

y = 1 + I(x>=0) -(x-1)\*I(x>=1)

plot(x,y,type="b")

截图：



5.问题（略）

（a）问题（略）

g2的训练RSS更小，其由于有着更高阶的导数，光滑度更高。

（b）问题（略）

g1的测试RSS更小，g2的高光滑度会造成过拟合的现象。

（c）问题（略）

当λ为0时，g1=g2，所以训练RSS和测试RSS都一样。

**应用**

9.问题（略）

（a）问题（略）

脚本：

set.seed(1)

library(MASS)

attach(Boston)

lm.fit = lm(nox ~ poly(dis, 3), data = Boston)

summary(lm.fit)

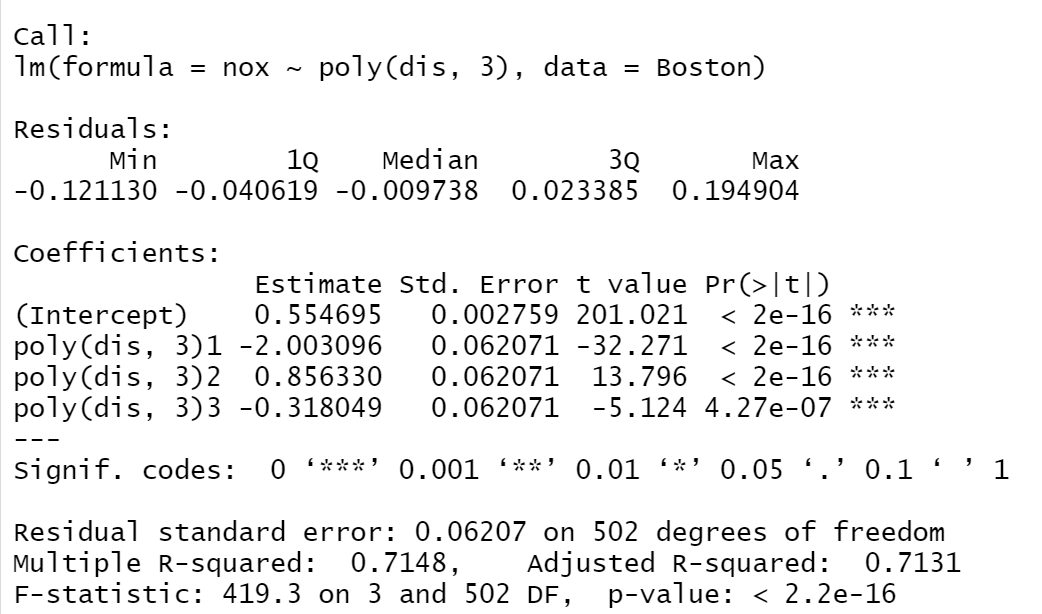
dislim = range(dis)

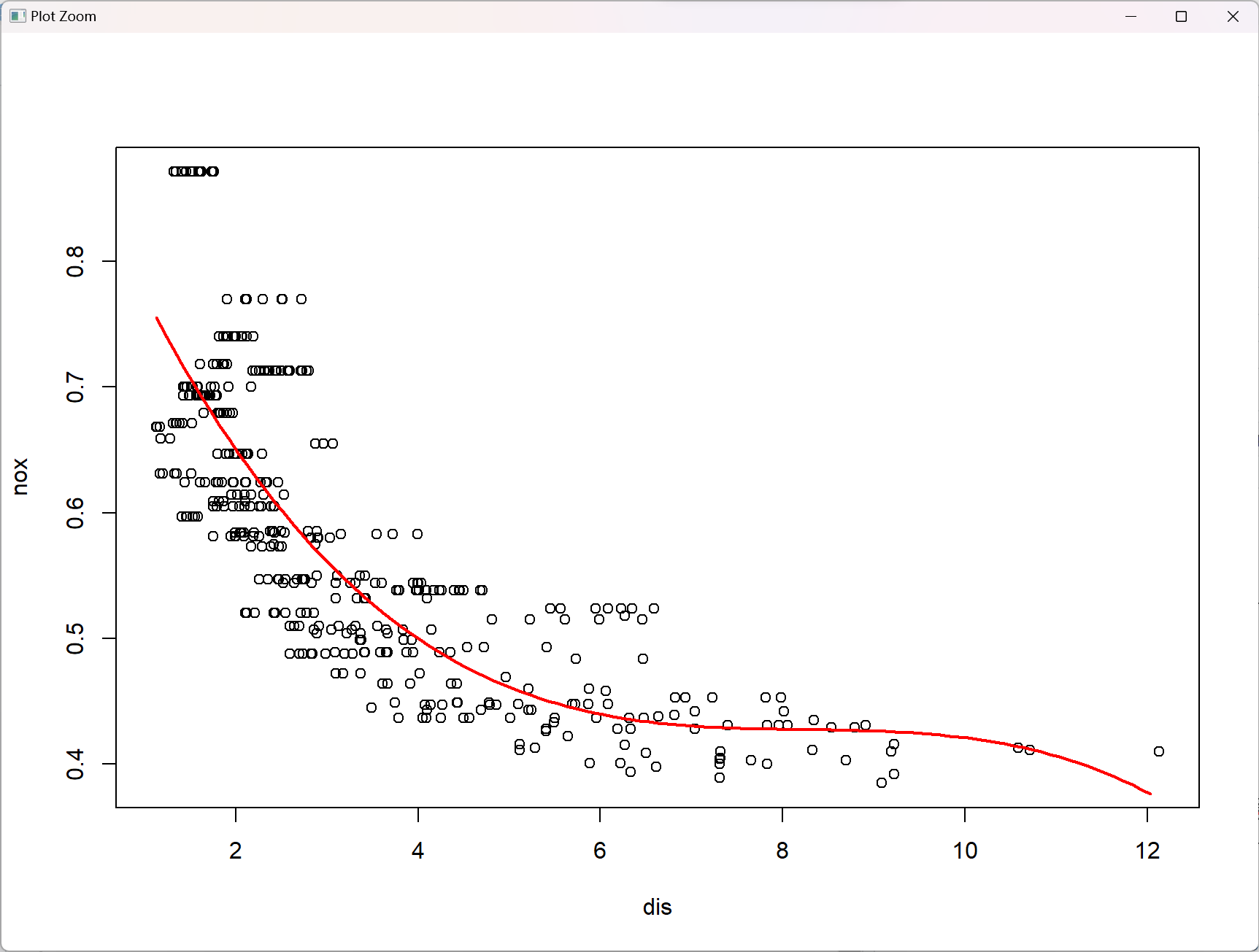
dis.grid = seq(from = dislim[1], to = dislim[2], by = 0.1)

lm.pred = predict(lm.fit, list(dis = dis.grid))

plot(nox ~ dis, data = Boston)

lines(dis.grid, lm.pred, col = "red", lwd = 2)

截图：  




由图所示，有着较好的拟合效果。

（b）问题（略）

脚本：

all.rss = rep(NA, 10)

par(mfrow=c(2,5))

for (i in 1:10) {

lm.fit = lm(nox ~ poly(dis, i), data = Boston)

all.rss[i] = sum(lm.fit$residuals^2)

lm.pred = predict(lm.fit, list(dis = dis.grid))

plot(nox ~ dis, data = Boston, col = "darkgrey")

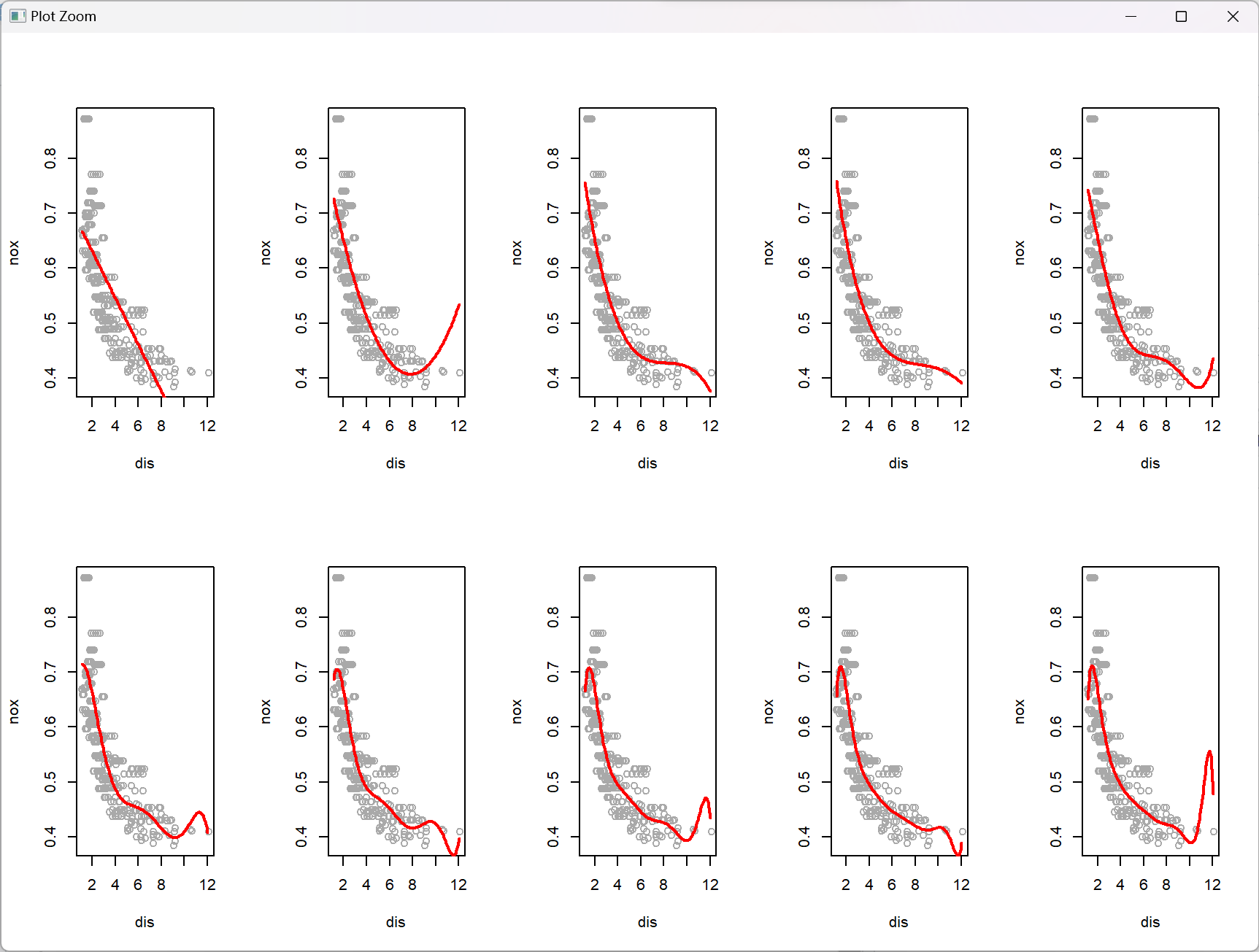
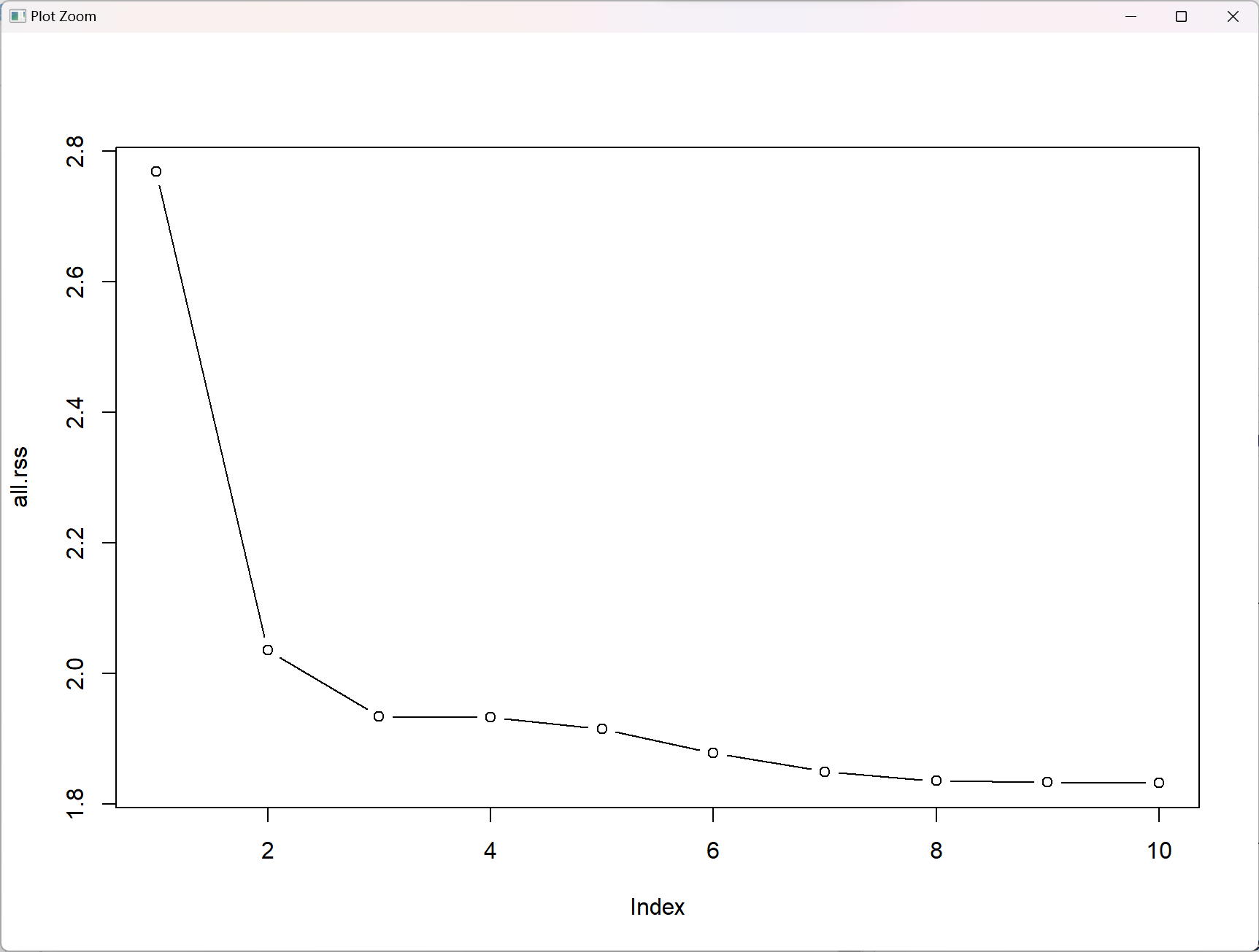
lines(dis.grid, lm.pred, col = "red", lwd = 2)

}

par(mfrow=c(1,1))

plot(all.rss, type="b")

截图：

右图所示，残差平方和随着维数增高而降低。

（c）问题（略）

脚本：

library(boot)

all.deltas = rep(NA, 10)

for (i in 1:10) {

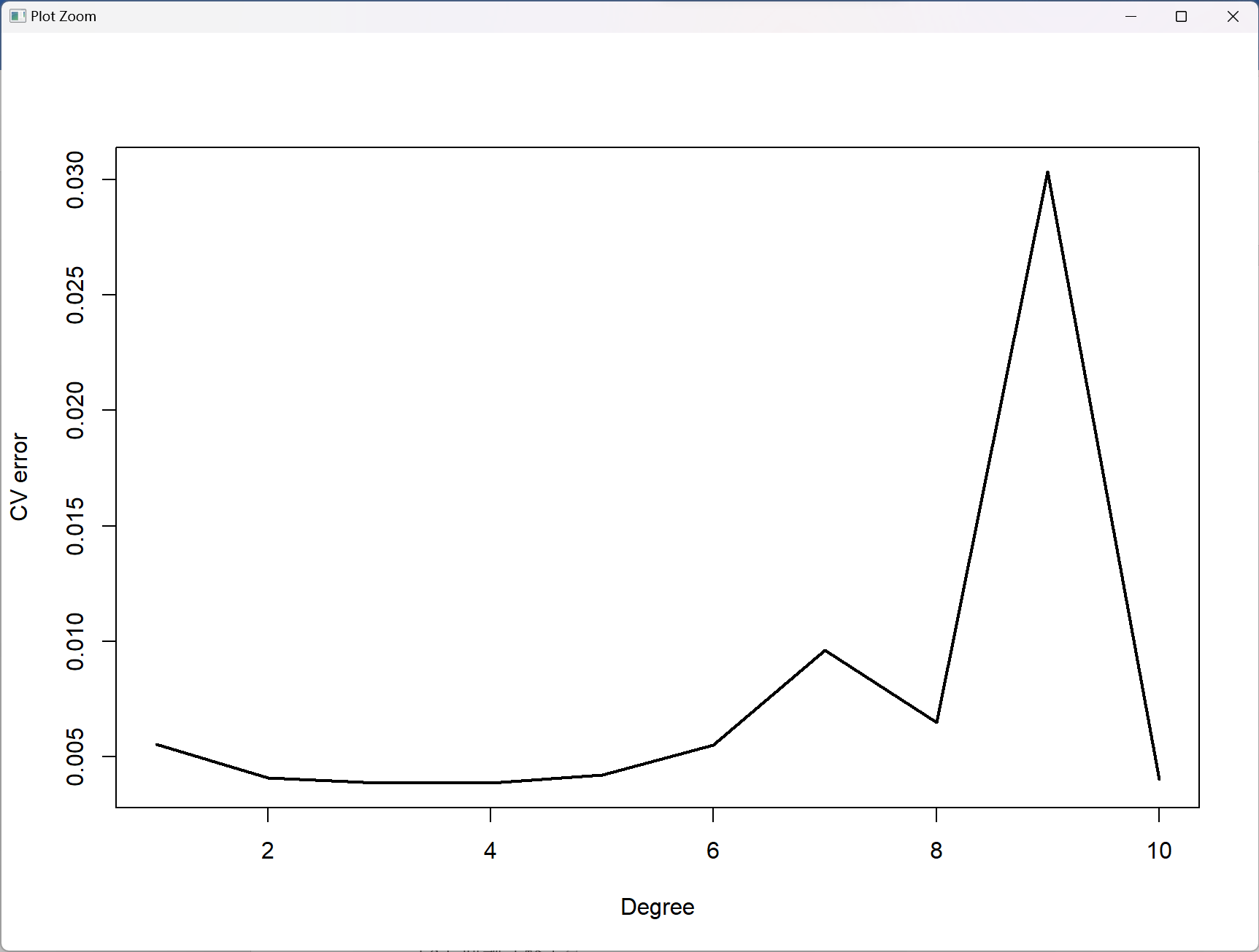
glm.fit = glm(nox ~ poly(dis, i), data = Boston)

all.deltas[i] = cv.glm(Boston, glm.fit, K = 10)$delta[2]

}

plot(1:10, all.deltas, xlab = "Degree", ylab = "CV error", type = "l", pch = 20, lwd = 2)

截图：



如图所示，在自由度为4的时候，交叉验证的误差最小。

（d）问题（略）

脚本：

range(dis)

library(splines)

sp.fit = lm(nox ~ bs(dis, df = 4, knots = c(4, 7, 10)), data = Boston)

summary(sp.fit)

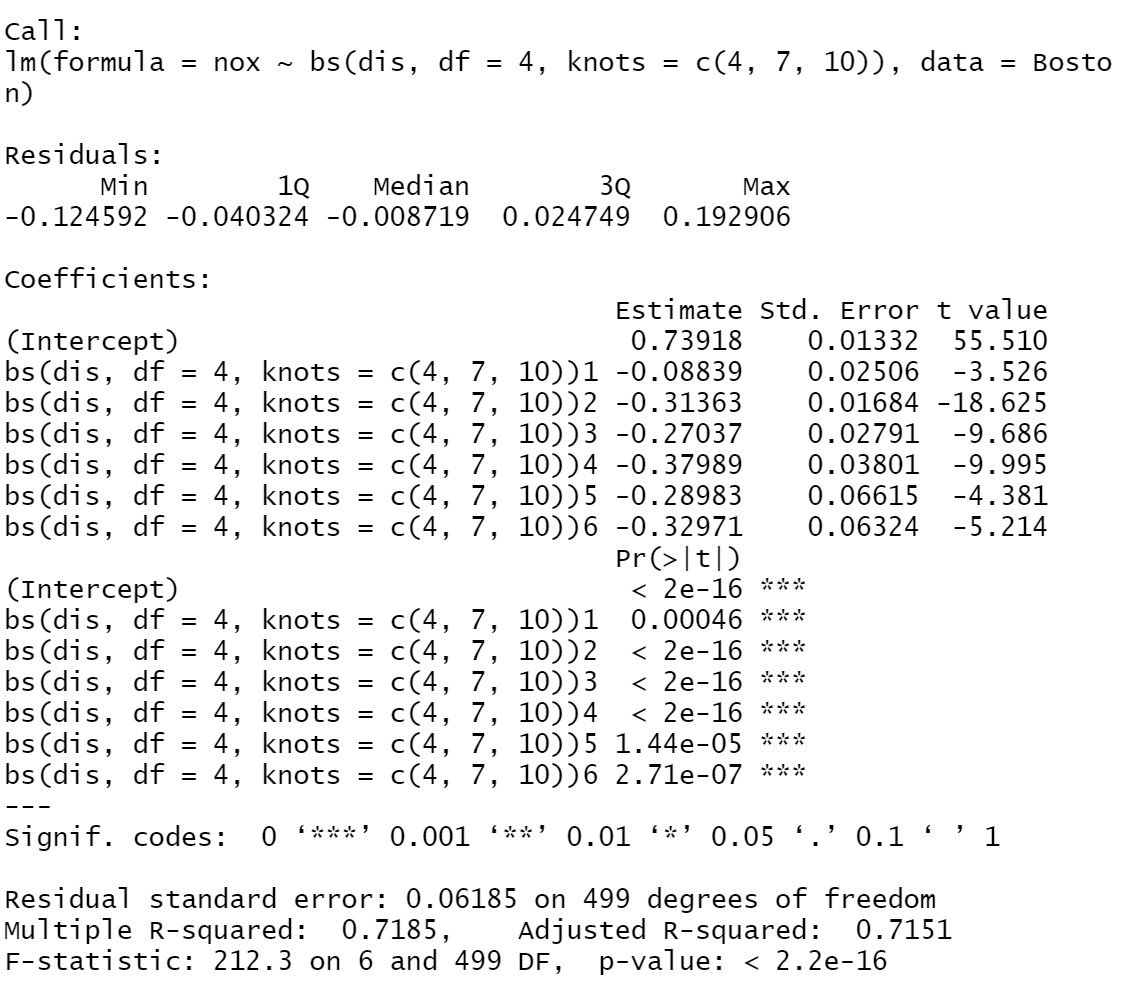
sp.pred = predict(sp.fit, list(dis = dis.grid))

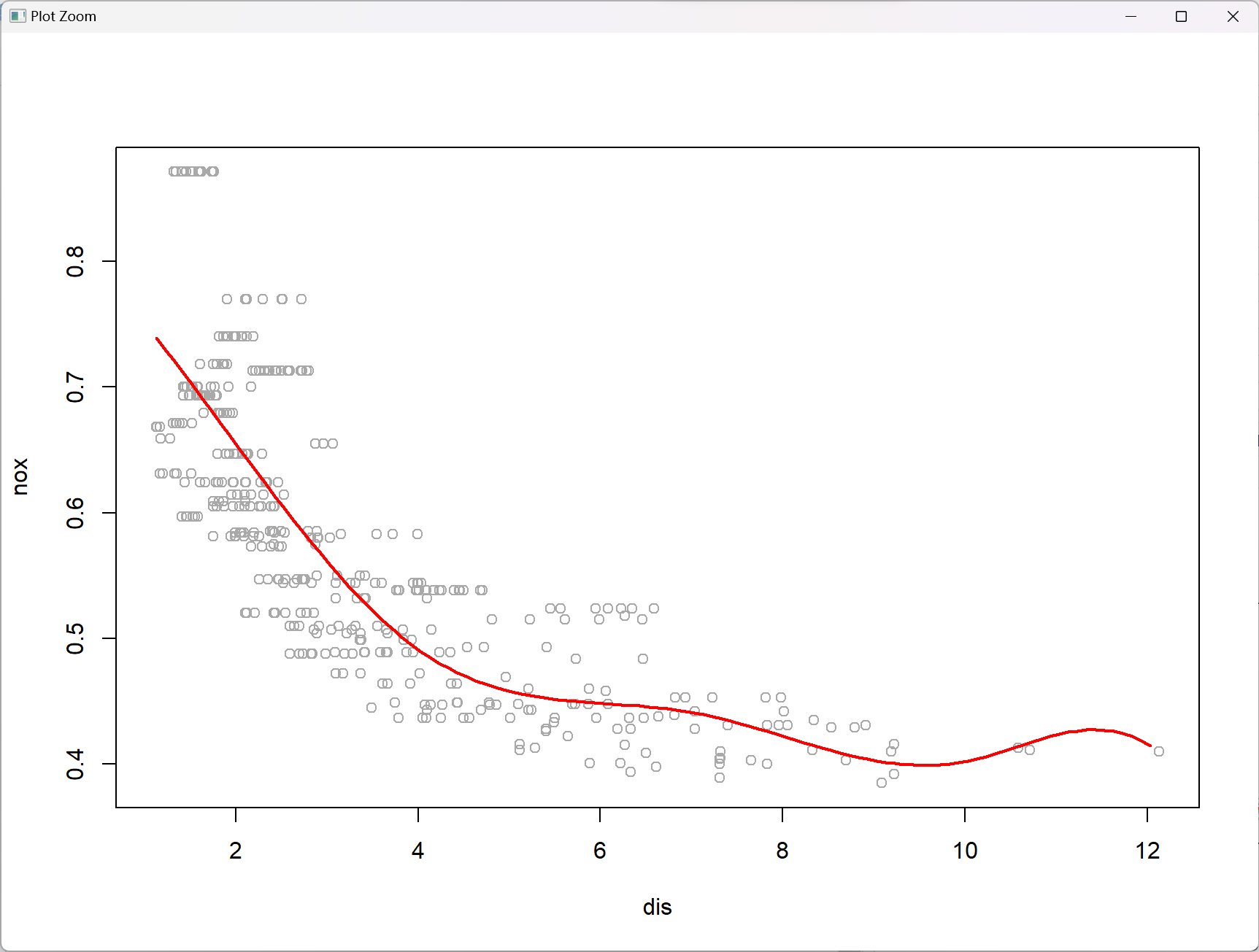
plot(nox ~ dis, data = Boston, col = "darkgrey")

lines(dis.grid, sp.pred, col = "red", lwd = 2)

截图：







自由度为4，选择三个节点，dis的范围在1到12，分成4个间隔，选取节点为4，7，10。

（e）问题（略）

脚本：

par(mfrow=c(4,4))

all.cv = rep(NA, 16)

for (i in 3:16) {

sp.fit = lm(nox ~ bs(dis, df = i), data = Boston)

all.cv[i] = sum(sp.fit$residuals^2)

sp.pred = predict(sp.fit, list(dis = dis.grid))

plot(nox ~ dis, data = Boston, col = "darkgrey")

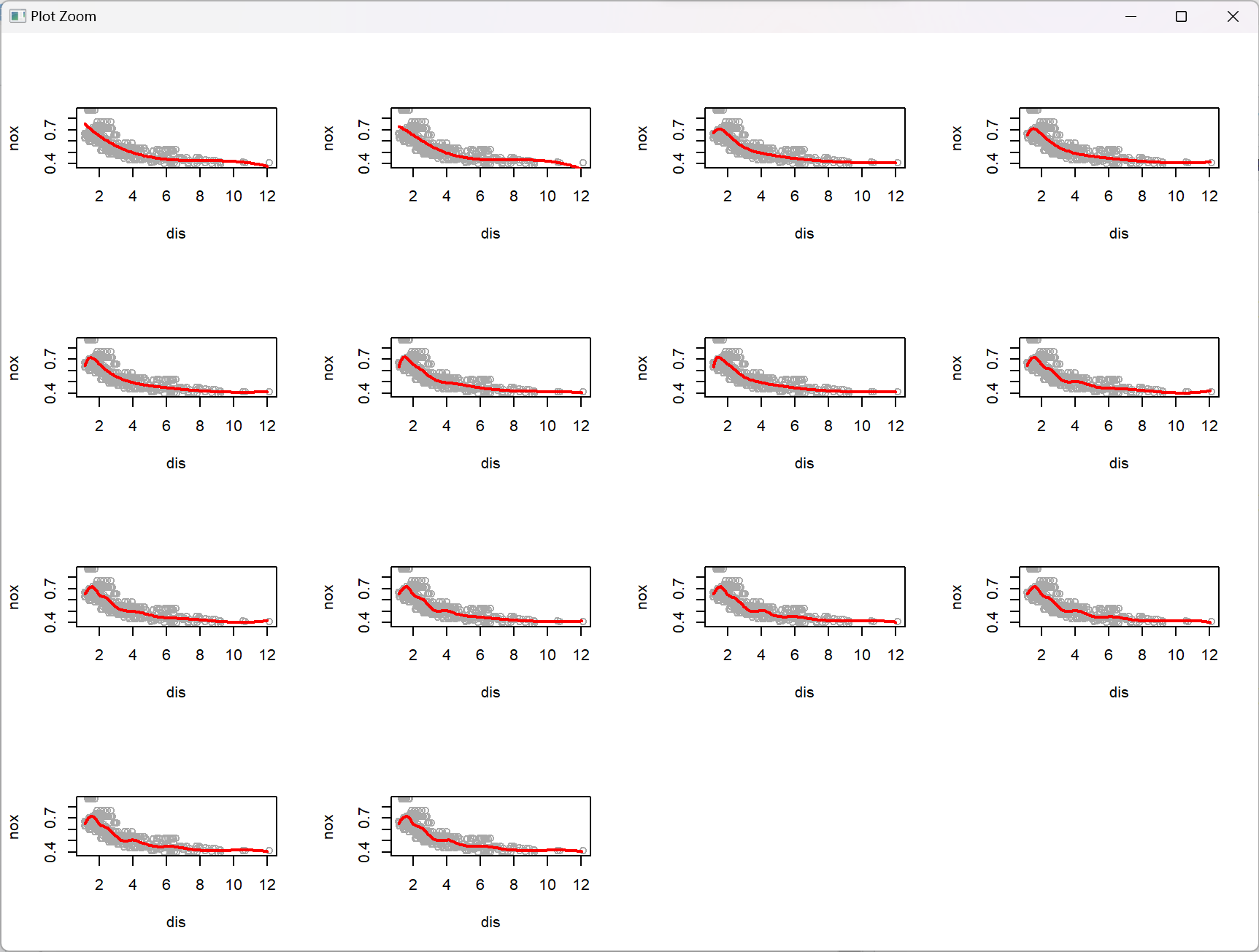
lines(dis.grid, sp.pred, col = "red", lwd = 2)

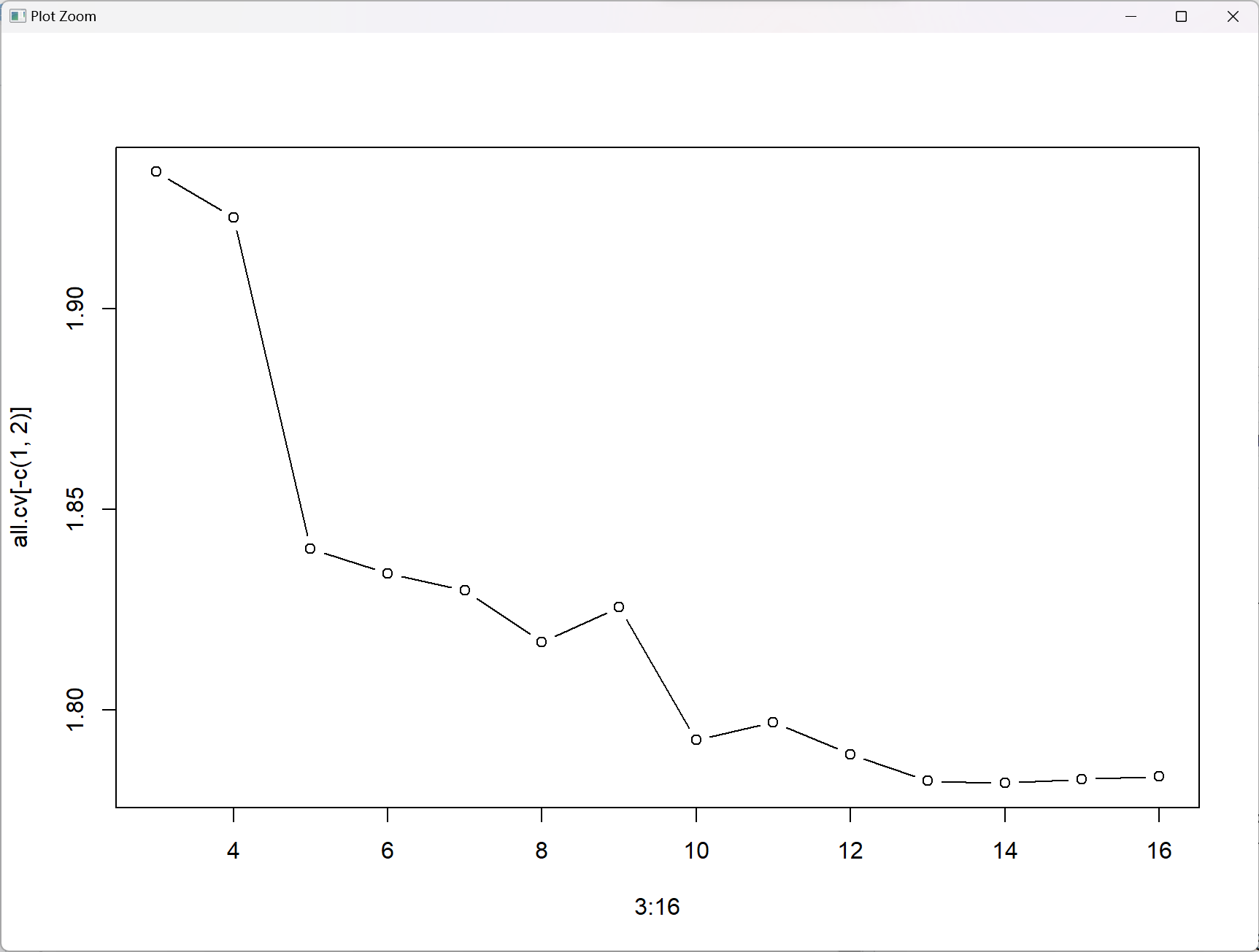
}

par(mfrow=c(1,1))

plot(3:16, all.cv[-c(1, 2)],type = "b")

截图：





如图所示，在自由度为13时，有最小的RSS。

（f）问题（略）

脚本：

all.cv = rep(NA, 16)

for (i in 3:16) {

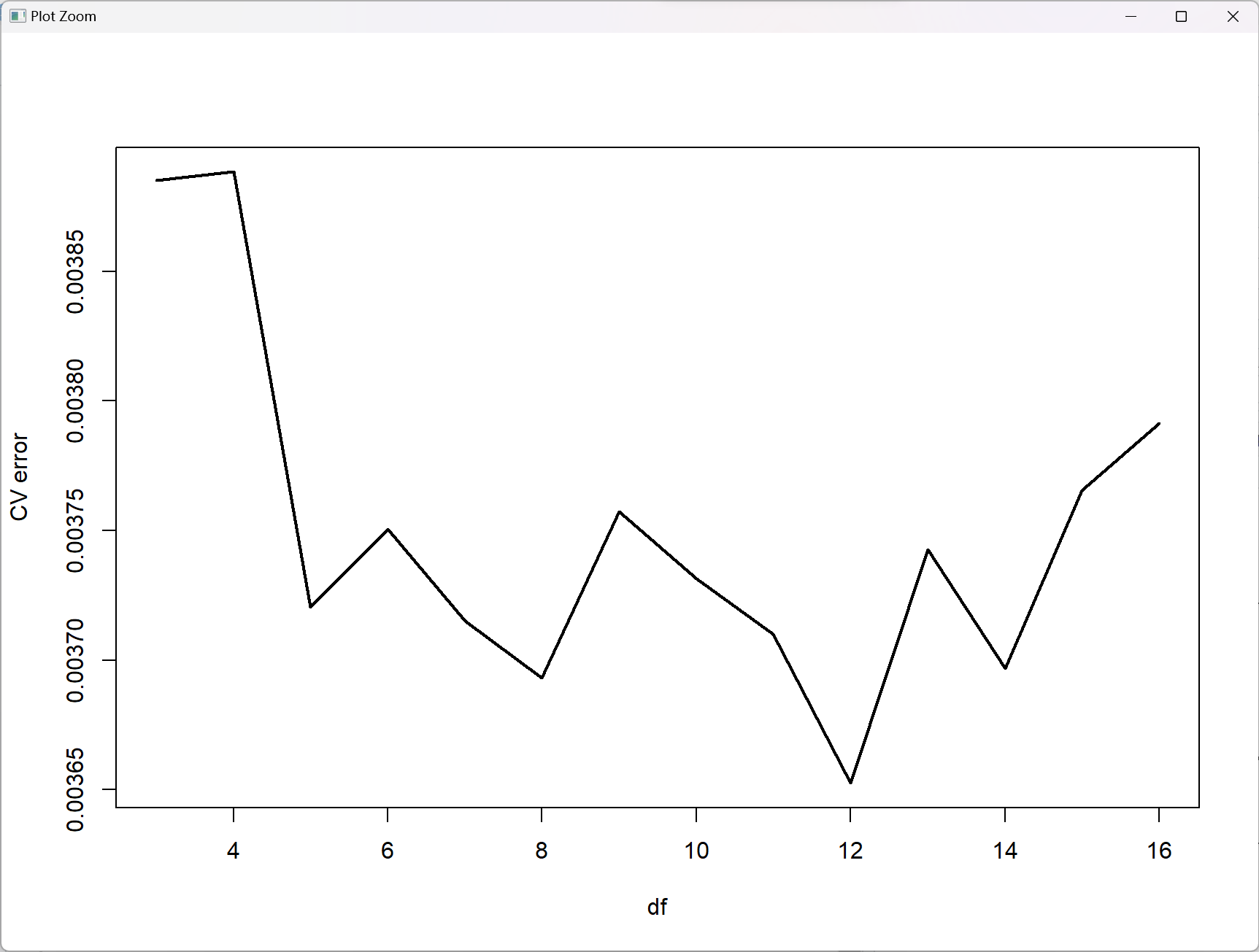
lm.fit = glm(nox ~ bs(dis, df = i), data = Boston)

all.cv[i] = cv.glm(Boston, lm.fit, K = 10)$delta[2]

}

plot(3:16, all.cv[-c(1, 2)], lwd = 2, type = "l", xlab = "df", ylab = "CV error")

截图：



如图所示，在十折交叉验证下，在自由度12的时候取得最小的误差。

10.问题（略）

（a）问题（略）

脚本：

set.seed(1)

library(ISLR)

library(leaps)

attach(College)

train = sample(length(Outstate), length(Outstate)/2)

test = -train

College.train = College[train, ]

College.test = College[test, ]

reg.fit = regsubsets(Outstate ~ ., data = College.train, nvmax = 17, method = "forward")

regfit.summary = summary(reg.fit)

which.min(regfit.summary$cp)

which.min(regfit.summary$bic)

which.max(regfit.summary$adjr2)

par(mfrow=c(1,3))

plot(regfit.summary$cp, xlab = "Subset Size", ylab = "Cp", type = "l")

plot(regfit.summary$bic, xlab = "Subset Size", ylab = "bic", type = "l")

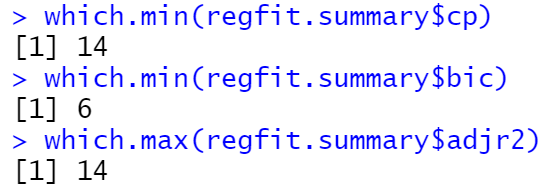
plot(regfit.summary$adjr2, xlab = "Subset Size", ylab = "adjr2", type = "l")

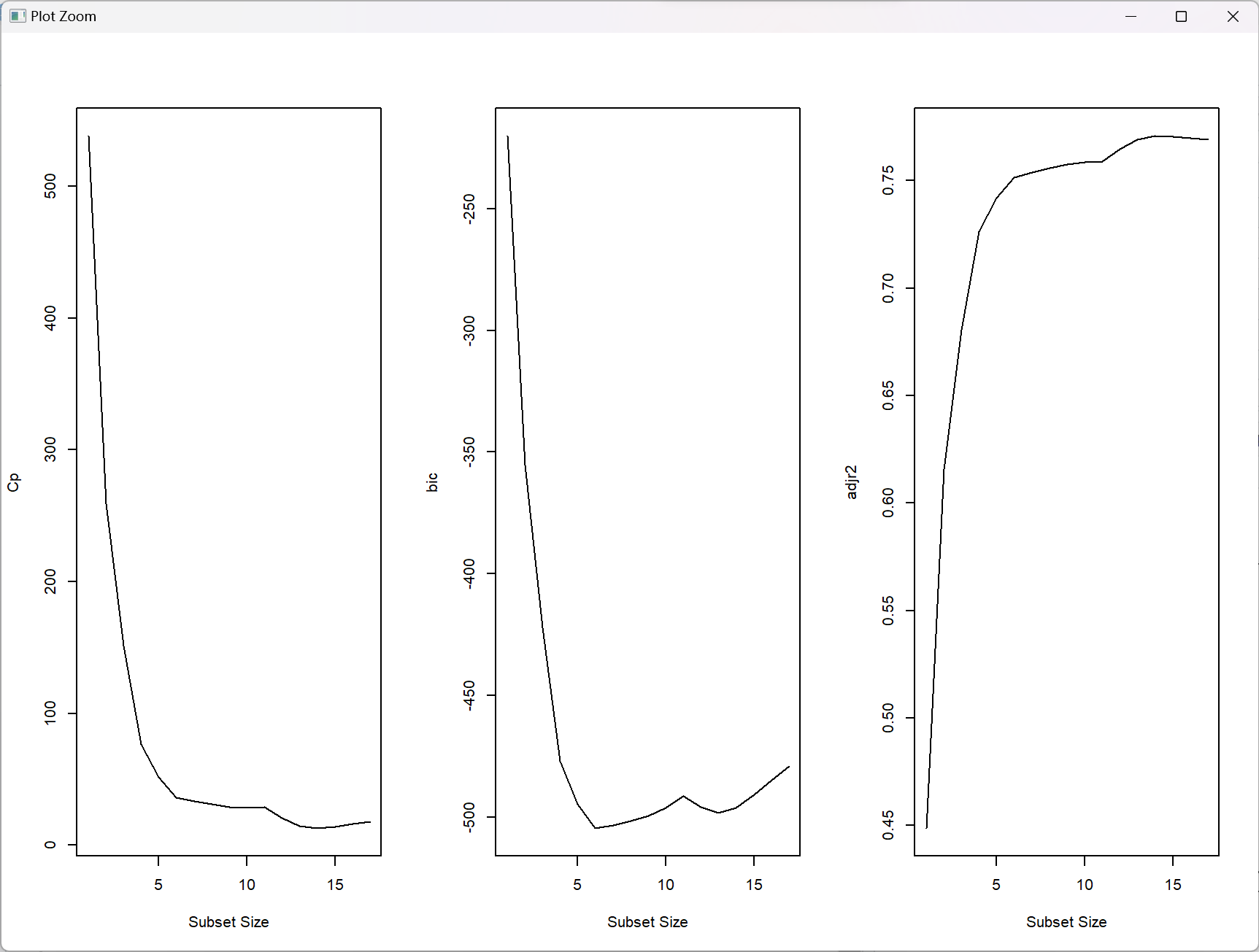
reg.fit = regsubsets(Outstate ~ ., data = College, method = "forward")

coefi = coef(reg.fit, id = 6)

names(coefi)

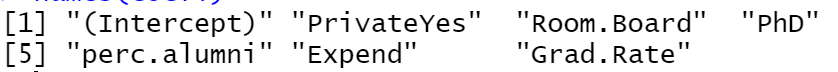
截图：





对于Cp和调整R方是14占优，而bic是6占优，根据图像可以得到，在三个图像中6和14的差距不大，为计算方便，采用6变量模型。

得到的6个变量为：



（b）问题（略）

脚本：

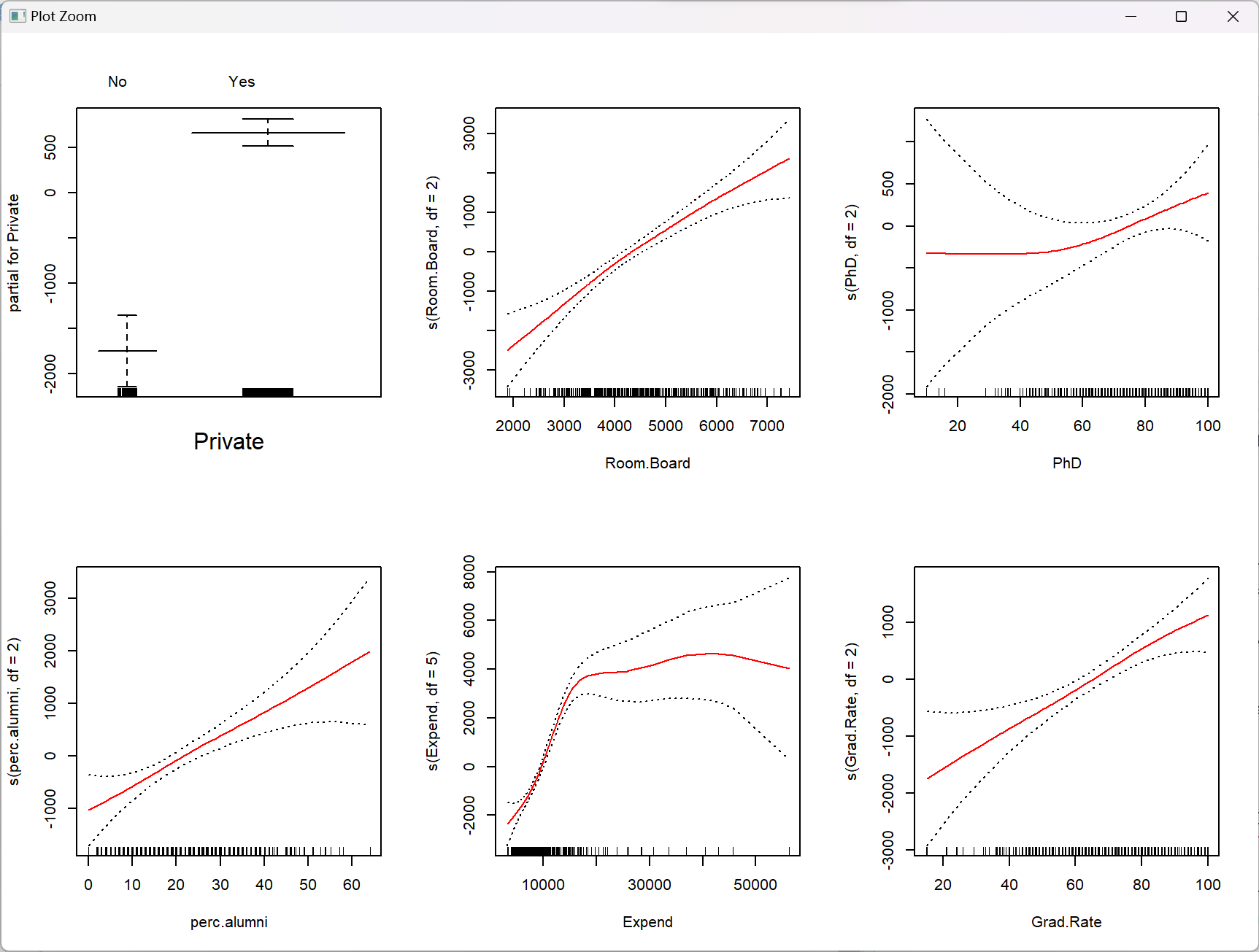
library(gam)

gam.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(gam.fit, se = T, col = "red")

截图：



如图所示，选出来的变量都与响应变量有较好的相关性。

（c）问题（略）

脚本：

gam.pred = predict(gam.fit, College.test)

gam.err = mean((College.test$Outstate - gam.pred)^2)

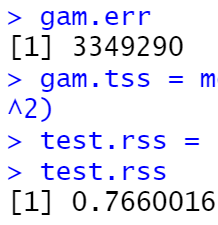
gam.err

gam.tss = mean((College.test$Outstate - mean(College.test$Outstate))^2)

test.rss = 1 - gam.err/gam.tss

test.rss

截图：



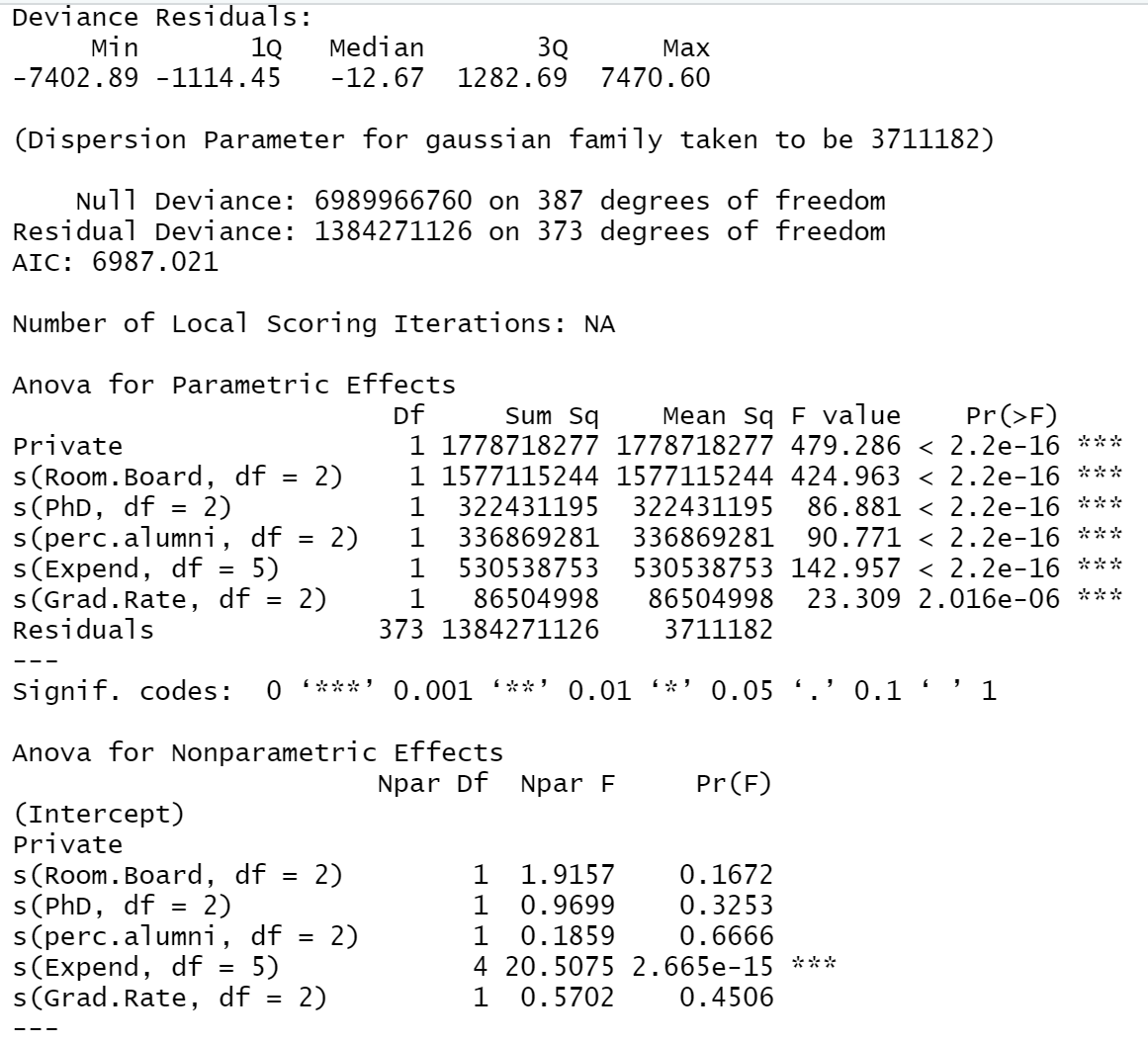
测试RSS为3349290，R方为0.766

（d）问题（略）

脚本：

summary(gam.fit)

截图：



通过观察p值，Expend的p值额外小，并根据（b）中图像，可以得出其与响应变量有显著的非线性关系，

11.问题（略）

（a）问题（略）

脚本：

set.seed(1)

X1 = rnorm(100)

X2 = rnorm(100)

eps = rnorm(100, sd = 0.1)

Y = -3 + 2 \* X1 - 0.5 \* X2 + eps

（b）问题（略）

脚本：

beta0 = rep(NA, 1000)

beta1 = rep(NA, 1000)

beta2 = rep(NA, 1000)

beta1[1] = 1

（c）问题（略）

见（e）

（d）问题（略）

见（e）

（e）问题（略）

脚本：

for (i in 1:1000) {

a = Y - beta1[i] \* X1

beta2[i] = lm(a ~ X2)$coef[2]

a = Y - beta2[i] \* X2

lm.fit = lm(a ~ X1)

if (i < 1000) {

beta1[i + 1] = lm.fit$coef[2]

}

beta0[i] = lm.fit$coef[1]

}

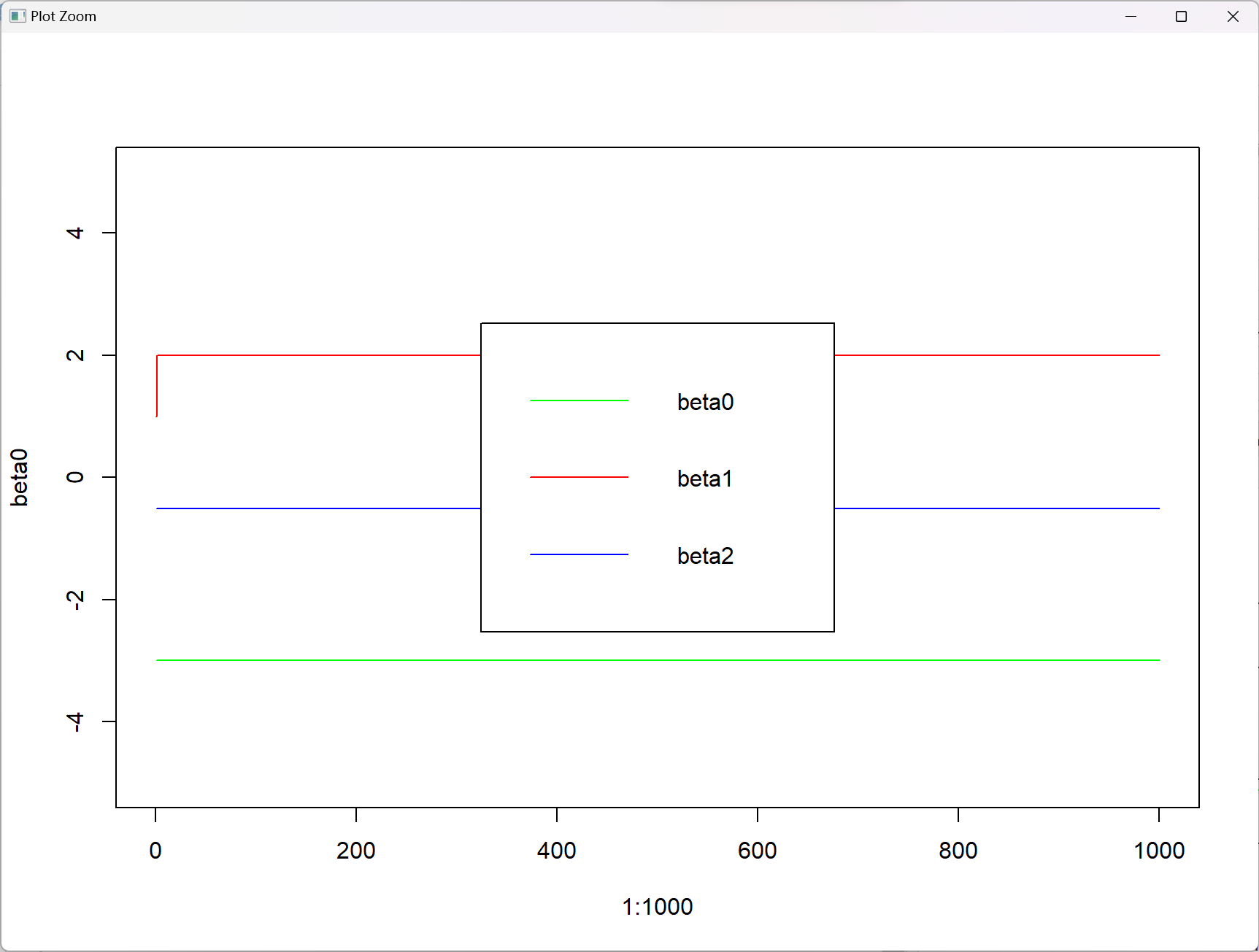
plot(1:1000, beta0, type = "l", ylim = c(-5, 5), col = "green")

lines(1:1000, beta1, col = "red")

lines(1:1000, beta2, col = "blue")

legend("center", c("beta0", "beta1", "beta2"), lty = 1, col = c("green", "red", "blue"))

截图：



如图所示，模型预测出的系数在很少的迭代之下便收敛到了真实值。

（f）问题（略）

脚本：

lm.fit = lm(Y ~ X1 + X2)

plot(1:1000, beta0, type = "l", xlab = "iteration", ylab = "betas", ylim = c(-5,5), col = "green")

lines(1:1000, beta1, col = "red")

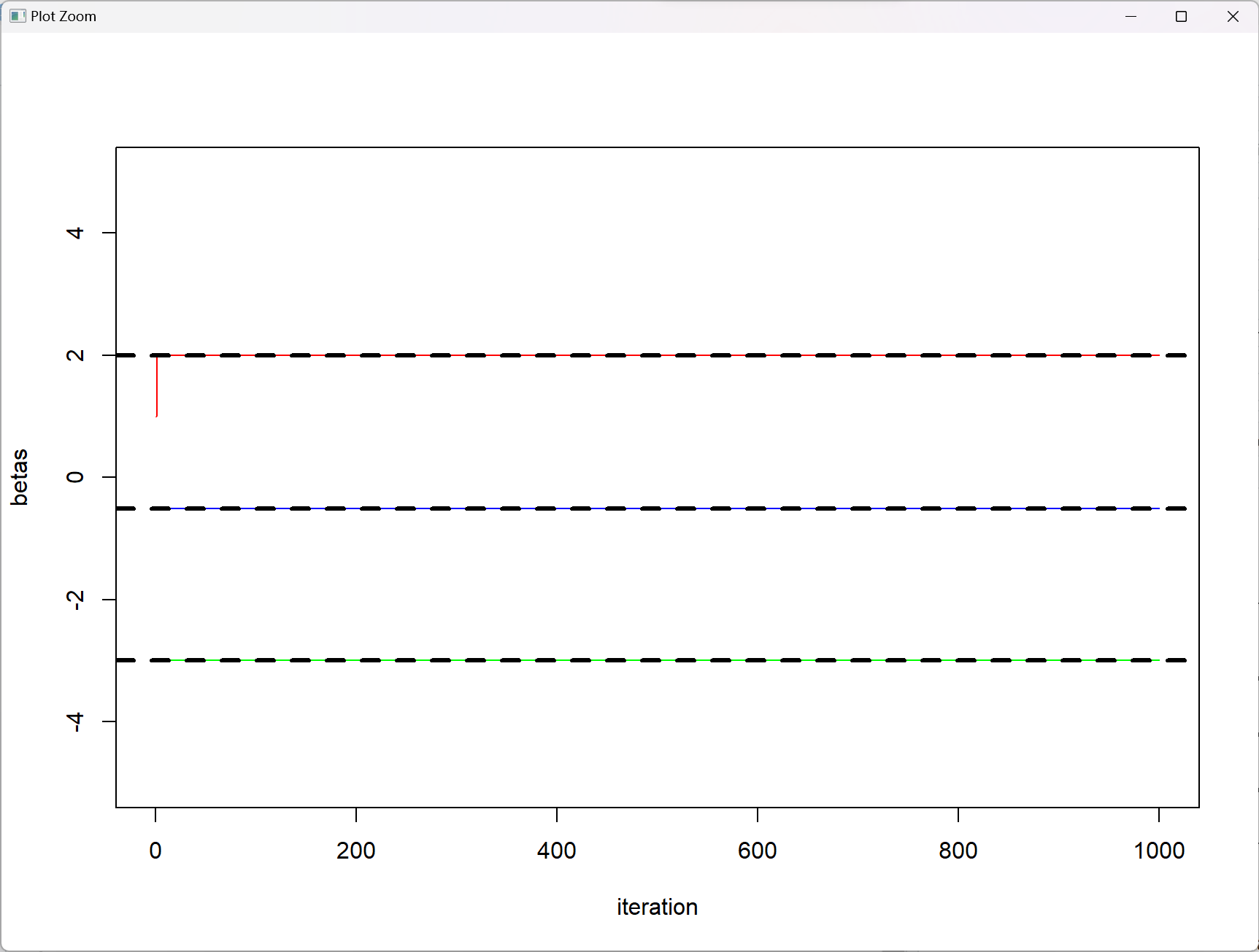
lines(1:1000, beta2, col = "blue")

abline(h = lm.fit$coef[1], lty = "dashed", lwd = 3)

abline(h = lm.fit$coef[2], lty = "dashed", lwd = 3)

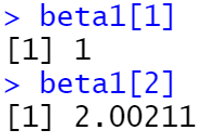
abline(h = lm.fit$coef[3], lty = "dashed", lwd = 3)

截图：



如图所示，得出的结果与（e）中完全相同。

（g）问题（略）



第一次迭代的前的beta1值是自设的，而一次迭代之后就接近于真实系数。所以对于线性关系，只需要一次迭代就能得到系数估计的近似结果。