HW11

M072040019 梅瀚中

第四題

(a)

iii. Steadily increase. 因為λ越大，β_j被限制的越多，模型越不靈活，training RSS 越大。

(b)

ii. Decrease initially, and then eventually start increasing in a U shape. 因為λ越大，β_j被限制的越多，模型越不靈活，一開始 test RSS 會下降，但過度不靈活的模型可能配適不佳，所以 test RSS 會漸漸增加。

(c)

iv. Steadily decrease. 因為λ越大，β_j被限制的越多，模型越不靈活，variance 會越小。

(d)

iii. Steadily increase. 因為λ越大，β_j被限制的越多，模型越部靈活，bias 會越大。

(e)

v. Remain constant. 因為λ越大，β_j被限制的越多，模型越部靈活，但 irreducible error 與模型不相關，所以 irreducible error 會保持常數。

第五題

(a)

minimize

$$(y_1 - \beta_0 - (\beta_1 x_{11} + \beta_2 x_{12}))^2 + (y_2 - \beta_0 - (\beta_1 x_{21} + \beta_2 x_{22}))^2 + \lambda(\beta_1^2 + \beta_2^2))$$

(b)

令$x_{11} = x_{12} = -x_{21} = -x_{22} = x$，$\beta_0 = 0$

$$S(\beta_1, \beta_2)$$
$$= (y_1 - \beta_0 - (\beta_1 x_{11} + \beta_2 x_{12}))^2 + (y_2 - \beta_0 - (\beta_1 x_{21} + \beta_2 x_{22}))^2 + \lambda(\beta_1^2 + \beta_2^2))$$
$$= (y_1 - x(\beta_1 + \beta_2))^2 + (y_2 + (\beta_1 + \beta_2))^2 + \lambda(\beta_1^2 + \beta_2^2))$$

對β_1, β_2偏微分，並令其為 0

$$\frac{\partial S(\beta_1, \beta_2)}{\partial \beta_1} = -2x(y_1 - x(\beta_1 + \beta_2)) + 2(y_2 + x(\beta_1 + \beta_2)) + 2\lambda\beta_1 = 0$$
$$\frac{\partial S(\beta_1, \beta_2)}{\partial \beta_1} = -2x(y_1 - x(\beta_1 + \beta_2)) + 2(y_2 + x(\beta_1 + \beta_2)) + 2\lambda\beta_2 = 0$$

得到

$$2\beta_1 x^2 + 2\beta_2 x^2 + \lambda\beta_1 = x(y_1 + y_2)$$
$$2\beta_1 x^2 + 2\beta_2 x^2 + \lambda\beta_2 = x(y_1 + y_2)$$

兩式相減

$$\lambda(\beta_1 - \beta_2) = 0$$

因此

$$\hat{\beta}_1 = \hat{\beta}_2$$

(c)

minimize

$$(y_1 - \beta_0 - (\beta_1 x_{11} + \beta_2 x_{12}))^2 + (y_2 - \beta_0 - (\beta_1 x_{21} + \beta_2 x_{22}))^2 + \lambda(|\beta_1| + |\beta_2|))$$

(d)

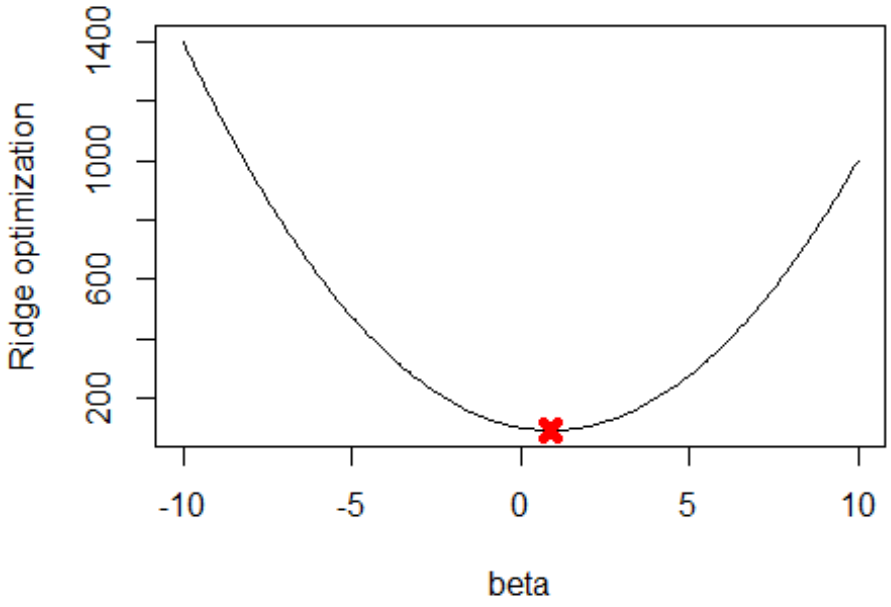
作法同(b)， 若$\beta_1\beta_2 > 0$，則$\hat{\beta}_1 = \hat{\beta}_2$ 若$\beta_1\beta_2 < 0$，則$\hat{\beta}_1 \neq \hat{\beta}_2$

第六題

(a)

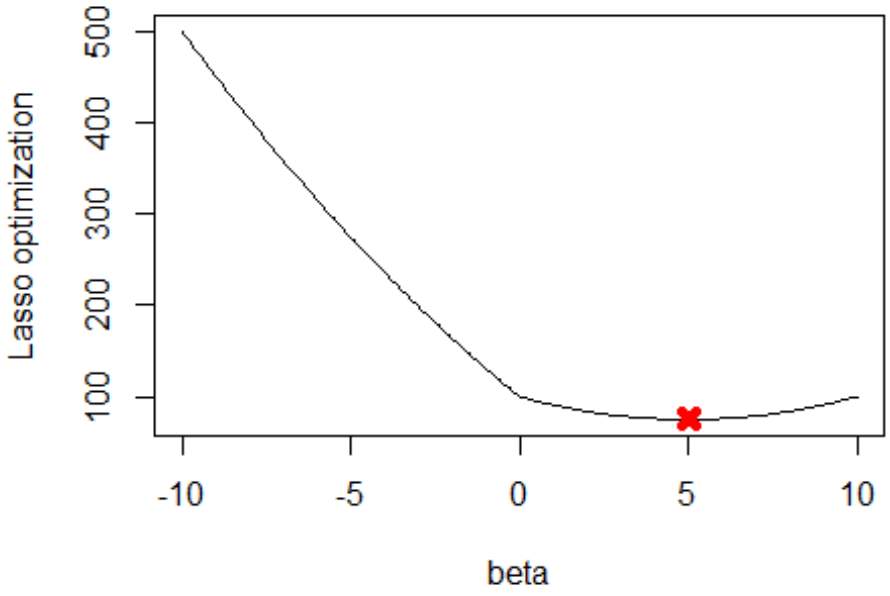
```
y <- 10
lambda <- 10
```

```r
beta <- seq(-10, 10, 0.1)
plot(beta, (y - beta)^2 + lambda * beta^2, type = "l", xlab = "beta", y
lab = "Ridge optimization")
beta.est <- y / (1 + lambda)
points(beta.est, (y - beta.est)^2 + lambda * beta.est^2, col = "red", p
ch = 4, lwd = 5)
```



(b)

```r
y <- 10
lambda <- 10
beta <- seq(-10, 10, 0.1)
plot(beta, (y - beta)^2 + lambda * abs(beta), type = "l", xlab = "beta"
, ylab = "Lasso optimization")
beta.est <- y - lambda / 2
points(beta.est, (y - beta.est)^2 + lambda * abs(beta.est), col = "red"
, pch = 4, lwd = 5)
```



第十題

(a)

```
set.seed(10)
x <- matrix(rnorm(1000 * 20), 1000, 20)
b <- rnorm(20)
z <- sample(1:20, 5)
b[z] <- 0
eps <- rnorm(1000)
y <- x %*% b + eps
```

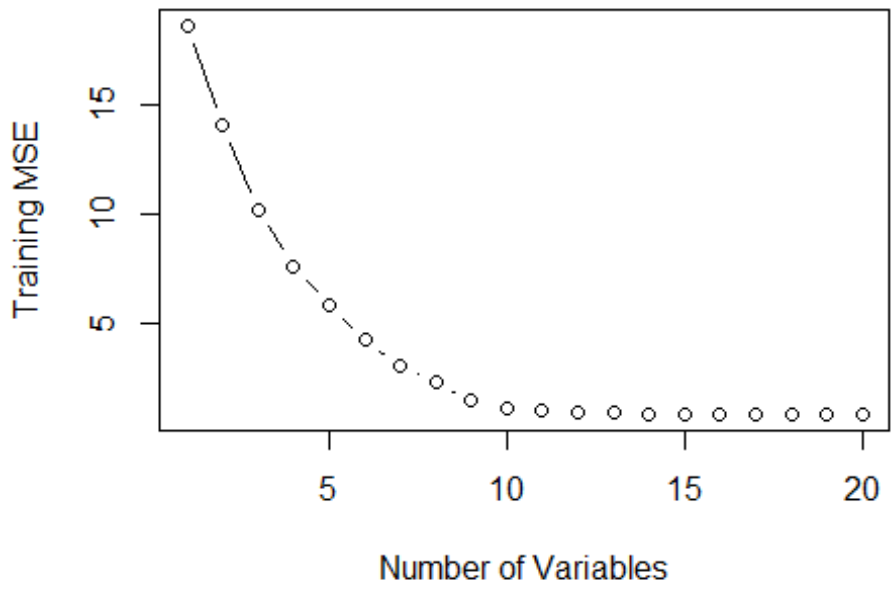
(b)

```
set.seed(10)
train <- sample(1:1000, 100)
x.train <- x[train,]
x.test <- x[-train,]
y.train <- y[train,]
y.test <- y[-train,]
```

(c)

```
library(leaps)
```

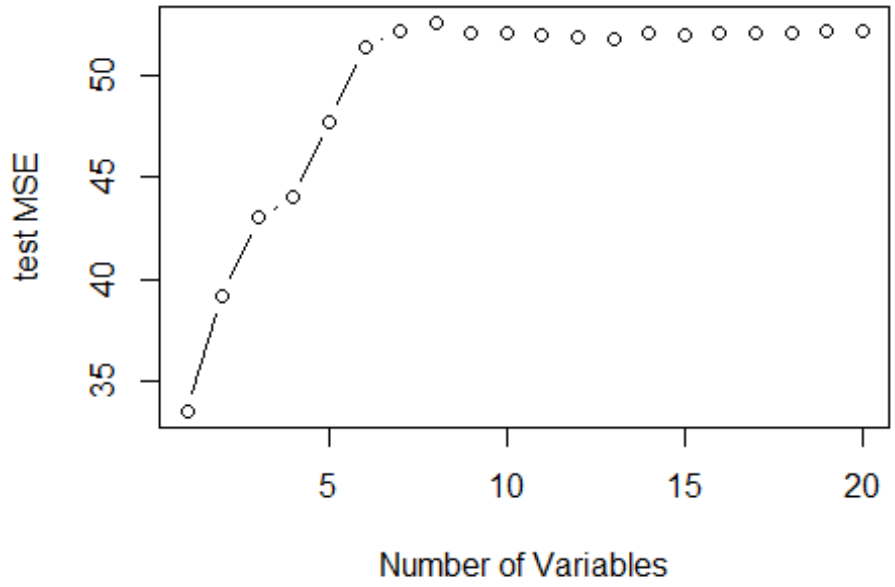
```
## Warning: package 'leaps' was built under R version 3.4.4

data.train <- data.frame(y = y.train, x = x.train)
regfit.full <- regsubsets(y~., data = data.train, nvmax = 20)
train.mat <- model.matrix(y ~ ., data = data.train, nvmax = 20)
val.errors <- rep(NA, 20)
for (i in 1:20) {
    coefi <- coef(regfit.full, id = i)
    pred <- train.mat[, names(coefi)] %*% coefi
    val.errors[i] <- mean((pred - y.train)^2)
}
plot(val.errors, xlab = "Number of Variables", ylab = "Training MSE", t
ype = "b")
```



(d)

```
library(leaps)
data.test <- data.frame(y = y.test, x = x.test)
regfit.full <- regsubsets(y~., data = data.train, nvmax = 20)
test.mat <- model.matrix(y ~ ., data = data.test, nvmax = 20)
val.errors <- rep(NA, 20)
for (i in 1:20) {
    coefi <- coef(regfit.full, id = i)
    pred <- test.mat[, names(coefi)] %*% coefi
    val.errors[i] <- mean((pred - y.train)^2)
}
```

```
plot(val.errors, xlab = "Number of Variables", ylab = "test MSE", type
= "b")
```



(e)

最小 test MSE 的變數個數

```
which.min(val.errors)

## [1] 1
```

(f)

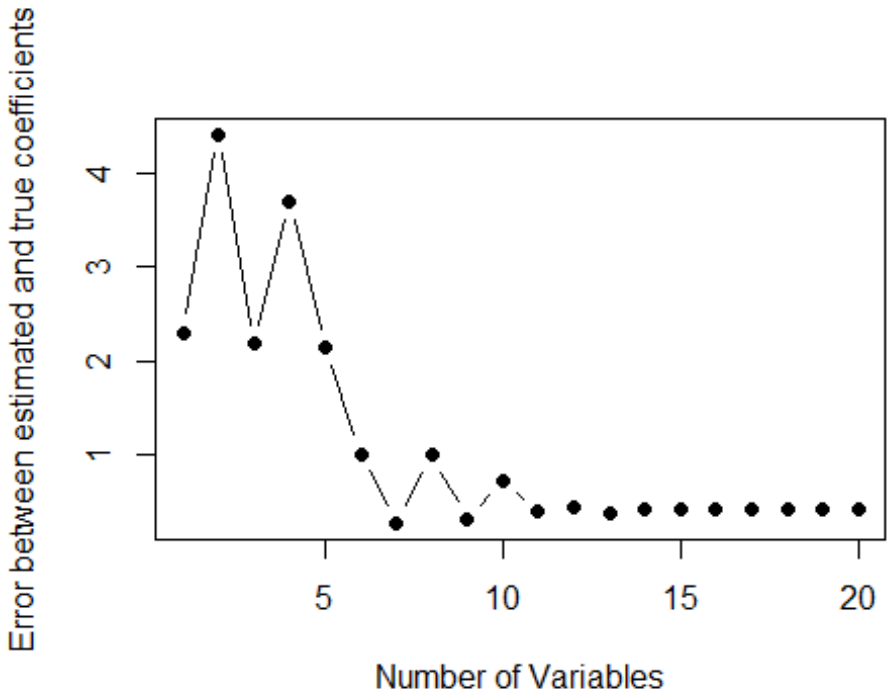
最小 test MSE 的模型係數

```
coef(regfit.full, which.min(val.errors))

## (Intercept)         x.13
##  -0.6273066   2.7254403
```

(g)

```
val.errors <- rep(NA, 20)
x_cols = colnames(x, do.NULL = FALSE, prefix = "x.")
for (i in 1:20) {
```

```
    coefi <- coef(regfit.full, id = i)
    val.errors[i] <- sqrt(sum((b[x_cols %in% names(coefi)] - coefi[name
s(coefi) %in% x_cols])^2) + sum(b[!(x_cols %in% names(coefi))])^2)
}
plot(val.errors, xlab = "Number of Variables", ylab = "Error between es
timated and true coefficients", pch = 19, type = "b")
```



真實與估計誤差差距最小值出現在變數個數為 7，而最小 test MSE 的模型係數為 1，表示模型配適越好並不表示有越小的 test MSE。

第十一題

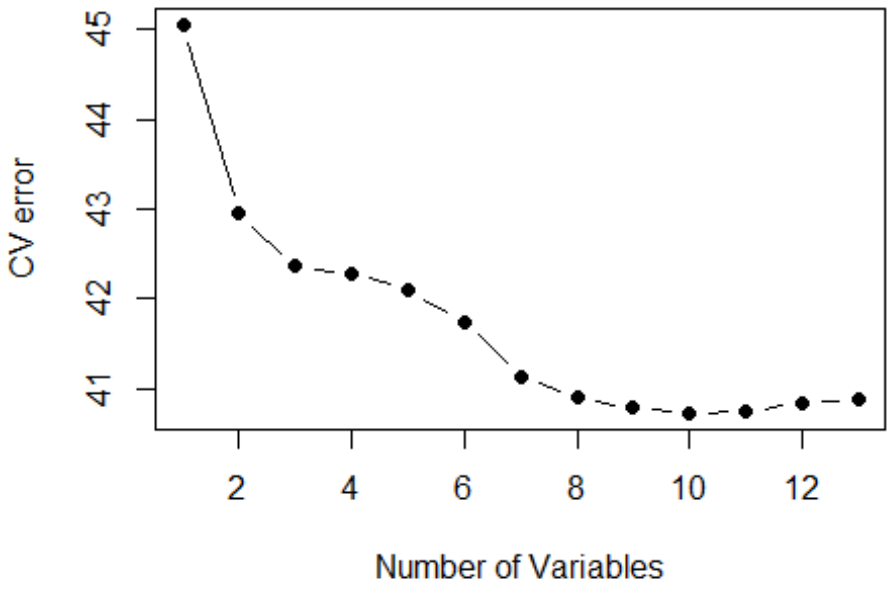
(a)

```
library(MASS)

## Warning: package 'MASS' was built under R version 3.4.4
```

i. best subset selection

```
library(leaps)
#使用交叉驗證方法
set.seed(10)
train <- sample(c(TRUE, FALSE), nrow(Boston), rep=TRUE)
regfit.best <- regsubsets(crim~., data = Boston[train,], nvmax = 13)

test.mat <- model.matrix(crim~., data = Boston[-train,])
```

```
val.errors <- rep(NA,13)
for (i in 1:13) {
  coefi = coef(regfit.best, id=i)
  pred = test.mat[,names(coefi)]%*%coefi
  val.errors[i] = mean((Boston$crim[-train]-pred)^2)
}

plot(val.errors, xlab = "Number of Variables", ylab = "CV error", pch =
 19, type = "b")
```



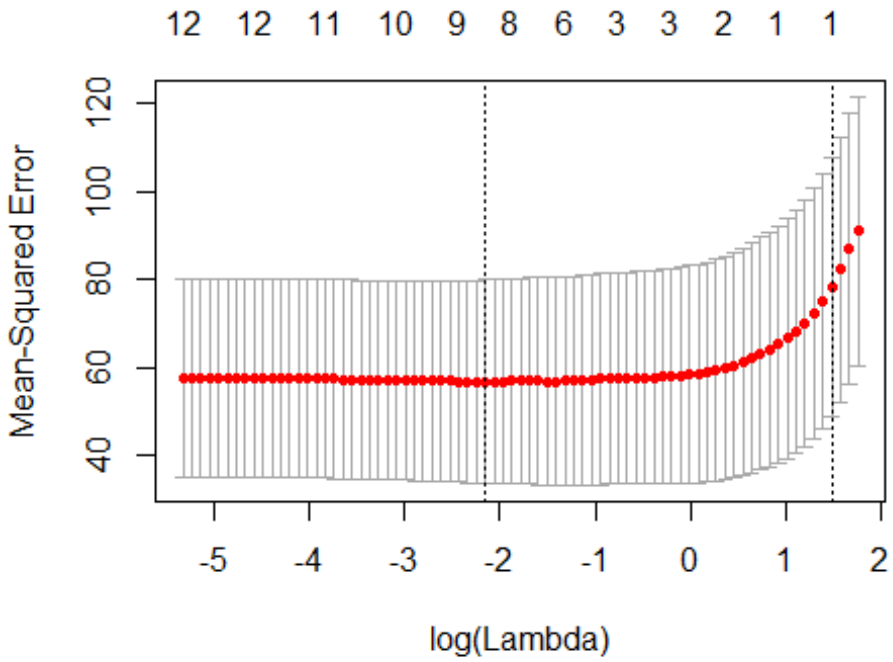
使用 best subset selection，當變數個數為 10 個時，有最低的 test error。 最小 test MSE

```
val.errors[which.min(val.errors)]

## [1] 40.72388
```

ii. lasso

```
library(glmnet)

## Warning: package 'glmnet' was built under R version 3.4.4

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.4.4

## Loaded glmnet 2.0-16
```

```
x <- model.matrix(crim~., Boston)[,-1]
y <- Boston$crim
grid <- 10^seq(10, -2, length = 100)
set.seed(10)
train <- sample(1:nrow(x), nrow(x)/2)
y.test <- y[-train]

lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
set.seed(10)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
plot(cv.out)
```



最佳λ

```
bestlam <- cv.out$lambda.min
bestlam

## [1] 0.1160257
```

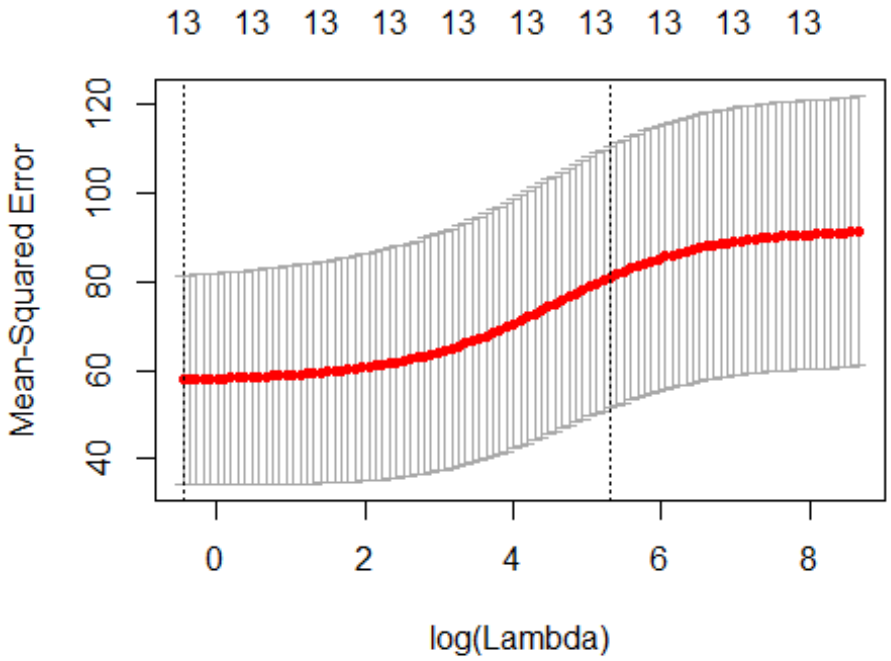
最小 test MSE

```
lasso.pred <- predict(lasso.mod, s=bestlam, newx = x[-train,])
mean((lasso.pred - y.test)^2)

## [1] 30.02633
```

iii. ridge

```
library(glmnet)
x <- model.matrix(crim~., Boston)[,-1]
```

```
y <- Boston$crim
grid <- 10^seq(10, -2, length = 100)
set.seed(10)
train <- sample(1:nrow(x), nrow(x)/2)
y.test <- y[-train]

ridge.mod <- glmnet(x[train,], y[train], alpha = 0, lambda = grid)
set.seed(10)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 0)
plot(cv.out)
```



最佳λ

```
bestlam <- cv.out$lambda.min
bestlam

## [1] 0.6337646
```

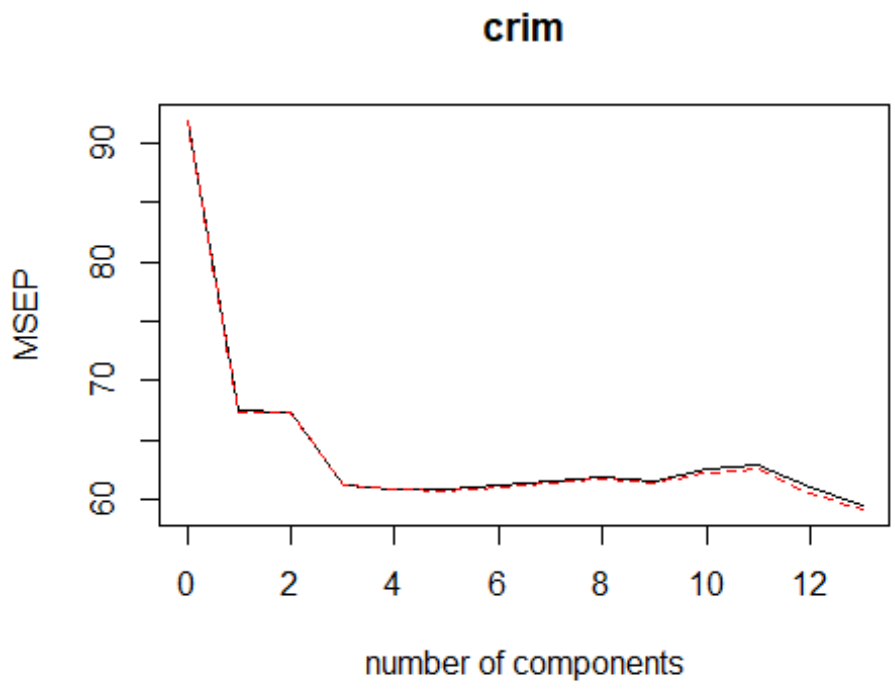
最小 test MSE

```
ridge.pred <- predict(ridge.mod, s=bestlam, newx = x[-train,])
mean((ridge.pred - y.test)^2)

## [1] 29.67505
```

iv. PCR

```
library(pls)

## Warning: package 'pls' was built under R version 3.4.4
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings

set.seed(10)
x <- model.matrix(crim~., Boston)[,-1]
y <- Boston$crim
train <- sample(1:nrow(x), nrow(x)/2)
test <- (-train)
y.test <- y[-train]
pcr.fit <- pcr(crim~., data = Boston, subset = train, scale = TRUE, val
idation = "CV")
validationplot(pcr.fit, val.type = "MSEP")
```



最小 test

MSE

```
pcr.pred <- predict(pcr.fit, x[test,], ncomp = 7)  #M=7 有最小 cross-vali
dation error
mean((pcr.pred - y.test)^2)  #test MSE

## [1] 32.18195
```

(b)

lasso 為最佳模型，因為其 test MSE 相對最小

(c)

的 lasso 模型係數

```
library(glmnet)
x <- model.matrix(crim~., Boston)[,-1]
y <- Boston$crim
grid <- 10^seq(10, -2, length = 100)
set.seed(10)
train <- sample(1:nrow(x), nrow(x)/2)
y.test <- y[-train]
lasso.mod <- glmnet(x[train,], y[train], alpha = 1, lambda = grid)
set.seed(1)
cv.out <- cv.glmnet(x[train,], y[train], alpha = 1)
bestlam <- cv.out$lambda.min
out <- glmnet(x,y,alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = "coefficient", s=bestlam)[1:14,]
lasso.coef
```

```
##  (Intercept)           zn        indus         chas          nox
##  8.772699444  0.030739105 -0.048484834 -0.503030668 -3.366471067
##           rm          age          dis          rad          tax
##  0.030902633  0.000000000 -0.577604215  0.492265525  0.000000000
##      ptratio        black        lstat         medv
## -0.097688218 -0.007545373  0.117936241 -0.122821380
```

所選的 lasso 模型沒有用到所有變數