HW5

賴冠維

2020/12/4

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

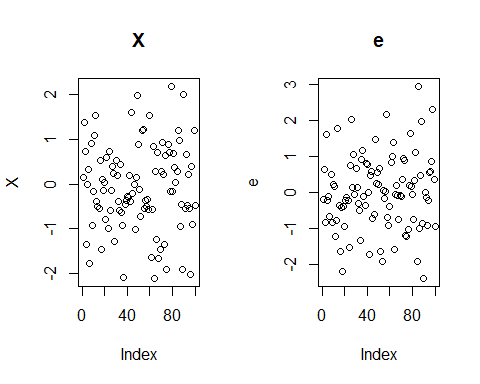
## √ ggplot2 3.3.2 √ purrr 0.3.4  
## √ tibble 3.0.4 √ dplyr 1.0.2  
## √ tidyr 1.1.2 √ stringr 1.4.0  
## √ readr 1.3.1 √ forcats 0.5.0

## Warning: package 'tibble' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

### (8)

#### (a)

設定set.seed(12345),以rnom()取出x,e各100個observations 

#### (b)

製造  
設定為

#### (c)

首先使用默認方法，即為Exhaustive Search (窮舉所有方法)，所得如下：  
可以看到默認為到擷取8個變數，並列出每個變數下表現最好的變數組合。

## Subset selection object  
## 10 Variables (and intercept)  
## Forced in Forced out  
## V1 FALSE FALSE  
## V2 FALSE FALSE  
## V3 FALSE FALSE  
## V4 FALSE FALSE  
## V5 FALSE FALSE  
## V6 FALSE FALSE  
## V7 FALSE FALSE  
## V8 FALSE FALSE  
## V9 FALSE FALSE  
## V10 FALSE FALSE  
## 1 subsets of each size up to 8  
## Selection Algorithm: exhaustive  
## V1 V2 V3 V4 V5 V6 V7 V8 V9 V10  
## 1 ( 1 ) " " " " "\*" " " " " " " " " " " " " " "  
## 2 ( 1 ) " " "\*" "\*" " " " " " " " " " " " " " "  
## 3 ( 1 ) "\*" "\*" "\*" " " " " " " " " " " " " " "  
## 4 ( 1 ) "\*" "\*" "\*" "\*" " " " " " " " " " " " "  
## 5 ( 1 ) "\*" "\*" "\*" " " " " "\*" " " " " " " "\*"  
## 6 ( 1 ) "\*" "\*" "\*" " " "\*" " " " " "\*" " " "\*"  
## 7 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" " " " "  
## 8 ( 1 ) "\*" "\*" "\*" "\*" "\*" "\*" " " "\*" " " "\*"

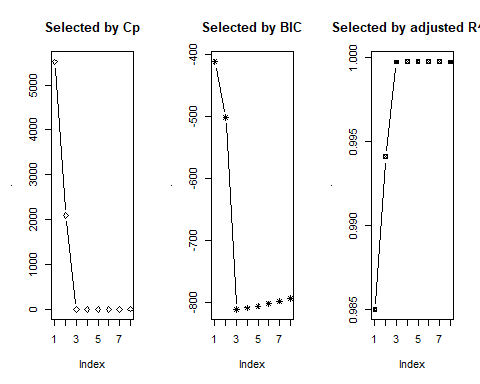
接著以三種不同標準來挑選變數，  
以Cp(複雜度)最低、BIC值最小、Adj R^2最大為標準進行變數篩選，  
可以看到這三種方法()分別挑選了4、3、5個變數，  
並且變數組成也不盡相同，代表不同方法所在意的地方都各有差異。

## [1] "Cp Select: 4 Variables"

## (Intercept) V1 V2 V3 V4   
## 0.9478419 9.5931787 5.2919897 20.1234912 -0.1000258

## [1] "BIC Select: 3 Variables"

## (Intercept) V1 V2 V3   
## 1.072448 9.610514 4.907571 20.117291



## [1] "Adj R^2 Select: 5 Variables"

## (Intercept) V1 V2 V3 V6 V10   
## 0.905780200 9.615345281 5.407379418 20.099445433 -0.077575224 0.002591922

#### (d)

首先使用Forward stepwise Selection，Forward Stepwise的作法：  
\* 在一個空的迴歸中逐一添加變數，直到任何一個變數的額外貢獻度(AIC、BIC、Cp等)無統計意義就停止。

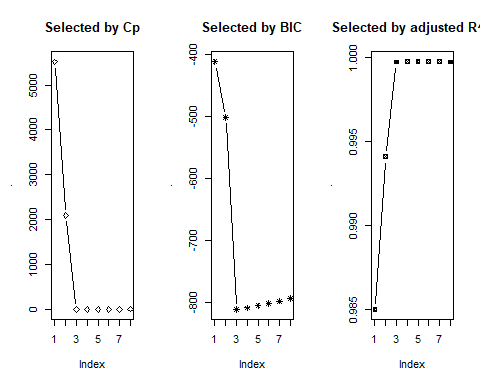
可以看到這三種方法()分別挑選了4、3、5個變數，  
與前述結果相同。

## [1] "Cp Select: 4 Variables"

## (Intercept) V1 V2 V3 V4   
## 0.9478419 9.5931787 5.2919897 20.1234912 -0.1000258

## [1] "BIC Select: 3 Variables"

## (Intercept) V1 V2 V3   
## 1.072448 9.610514 4.907571 20.117291



## [1] "Adj R^2: 5 Variables"

## (Intercept) V1 V2 V3 V6 V10   
## 0.905780200 9.615345281 5.407379418 20.099445433 -0.077575224 0.002591922

接下來採用Backwards stepwise Selection，Backward Stepwise：  
\* 在一個完整的迴歸中，逐一移除變數，直到移除任何一個變數時，模型都會損失過多的解釋力，那就停止。

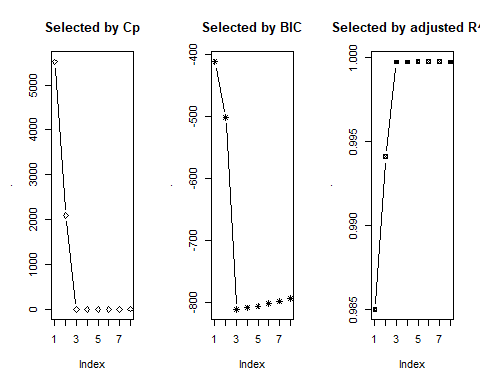
可以看到這三種方法()分別挑選了3、3、5個變數，  
僅Cp挑選結果改變，其餘相同。

## [1] "Cp Select: 3 Variables"

## (Intercept) V1 V2 V3   
## 1.072448 9.610514 4.907571 20.117291

## [1] "BIC Select: 3 Variables"

## (Intercept) V1 V2 V3   
## 1.072448 9.610514 4.907571 20.117291



## [1] "Adj R^2: 5 Variables"

## (Intercept) V1 V2 V3 V6 V10   
## 0.905780200 9.615345281 5.407379418 20.099445433 -0.077575224 0.002591922

#### (e)

使用Lasso Regression，並且使用Cross Validation 來挑選最佳的，可由下圖所見：  
不論是或是皆選取3個變數。

## Loading required package: glmnet

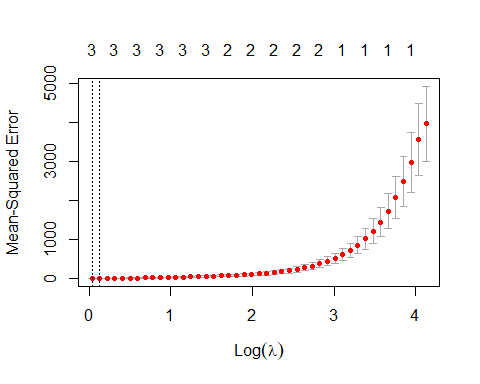
## Warning: package 'glmnet' was built under R version 4.0.3

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

## Loaded glmnet 4.0-2



##   
## Call: cv.glmnet(x = d[, -11], y = d[, 11], nfolds = 10, family = "gaussian", alpha = 1)   
##   
## Measure: Mean-Squared Error   
##   
## Lambda Measure SE Nonzero  
## min 1.036 4.714 1.364 3  
## 1se 1.137 5.387 1.662 3

選到，其參數為9.148515,3.866065,19,799337

## 10 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## V1 9.148515  
## V2 3.866065  
## V3 19.799337  
## V4 .   
## V5 .   
## V6 .   
## V7 .   
## V8 .   
## V9 .   
## V10 .

選到，其參數為9.105123,3.764425,19,767835

## 10 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## V1 9.105123  
## V2 3.764425  
## V3 19.767835  
## V4 .   
## V5 .   
## V6 .   
## V7 .   
## V8 .   
## V9 .   
## V10 .

可以發現不論是或是其所選取之變數以及所配飾參數的值 皆與使用Forward、Backward Selection時採用Cp、BIC標準時  
所選取之變數相同，配飾參數的值也相近。

#### (f)

製造新的

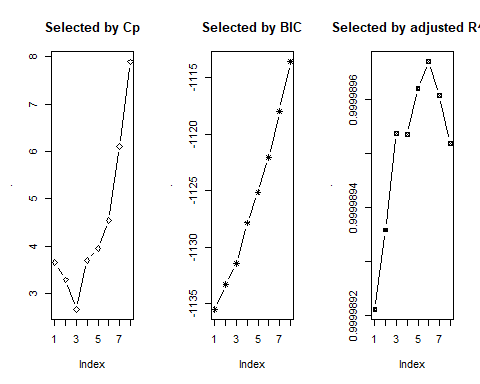
可以發現三種不同方法所選取的變數皆不同，相同的是皆選取了並且參數十分接近當初所模擬的值，  
可能是因為為7次方項，整個幾乎由這個變數決定，造成其餘變數估計較不準確， 但是BIC所選取變數與當初設定相同，而且配適參數相當接近，是三個當中表現最佳者。

## [1] "Cp Select: 3 Variables"

## (Intercept) V1 V6 V7   
## 5.048981502 -0.232936749 -0.008446317 7.004208175

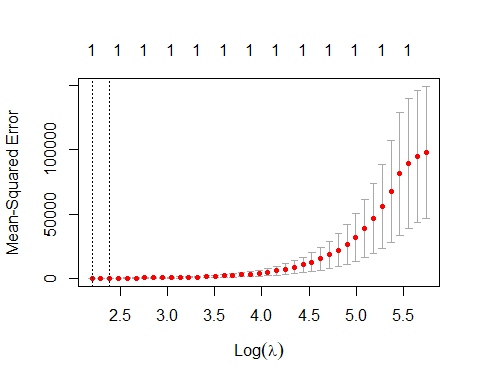
## [1] "BIC Select: 3 Variables"

## (Intercept) V7   
## 4.994720 7.001095



## [1] "Adj R^2: 5 Variables"

## (Intercept) V1 V3 V4 V6 V7   
## 4.94075112 -0.53434753 0.24072919 0.60752611 -0.36064537 6.99332263   
## V8   
## 0.04919629

由Lasso Regression所篩選之變數，皆僅選取出來  
可以發現Lasso Regression可能為更保守的變數選取方法 

##   
## Call: cv.glmnet(x = d[, -11], y = d[, 11], nfolds = 10, family = "gaussian", alpha = 1)   
##   
## Measure: Mean-Squared Error   
##   
## Lambda Measure SE Nonzero  
## min 9.052 122.5 70.84 1  
## 1se 10.903 176.5 102.83 1

## 10 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## V1 .   
## V2 .   
## V3 .   
## V4 .   
## V5 .   
## V6 .   
## V7 6.791399488  
## V8 .   
## V9 0.000998944  
## V10 .

## 10 x 1 sparse Matrix of class "dgCMatrix"  
## s0  
## V1 .   
## V2 .   
## V3 .   
## V4 .   
## V5 .   
## V6 .   
## V7 6.749653974  
## V8 .   
## V9 0.000940358  
## V10 .

### (10)

#### (a)

建立 、、、

## num [1:1000, 1:20] 0.1567 1.37381 0.73067 -1.3508 -0.00851 ...  
## X

## num [1:20, 1] 0.001 0.0001 0.001 13 0.004 0.007 15 16 0.005 17 ...  
## beta

## num [1:1000, 1] 27.3 -67 36.8 -51.8 -48.3 ...  
## Y

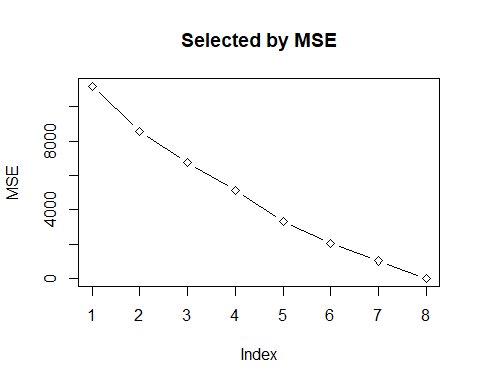
#### (b)

把DATA拆成Train、Test，Train有100筆、Test有900筆

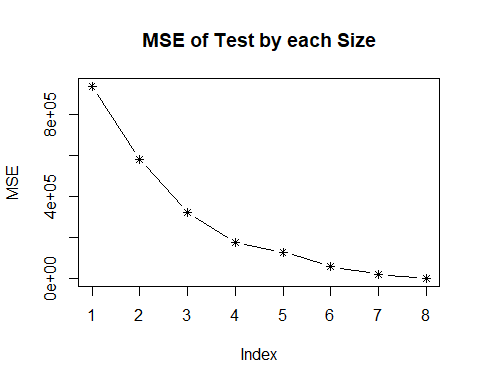
## Train: 100 21

## Test: 900 21

#### (c)

列出以Train Data配飾，在變數選取1個至8個時,表現最佳的模型，  
列出這8個模型的MSE，可以發現在增加變數個數後，MSE顯著遞減。 

#### (d)

用上述根據Train Data所選的所含變數1個至所含變數8個的最佳模型預測Test Data，  
一樣可以發現當變數數量增加，Test Data的MSE隨變數數量顯著遞減。 

#### (e)

由上圖可以看到在變數數量為8時有最小的MSE，因此認為是最佳的Model，  
下面列出所選的8個變數以及對Test Data預測時的MSE。

## [,1]  
## (Intercept) TRUE  
## X1 FALSE  
## X2 FALSE  
## X3 FALSE  
## X4 TRUE  
## X5 FALSE  
## X6 FALSE  
## X7 TRUE  
## X8 TRUE  
## X9 FALSE  
## X10 TRUE  
## X11 FALSE  
## X12 FALSE  
## X13 TRUE  
## X14 TRUE  
## X15 FALSE  
## X16 FALSE  
## X17 TRUE  
## X18 FALSE  
## X19 TRUE  
## X20 FALSE

## [1] "MSE of the Best Model in All Model Size(8 Variables): 113.087194280063"

#### (f)

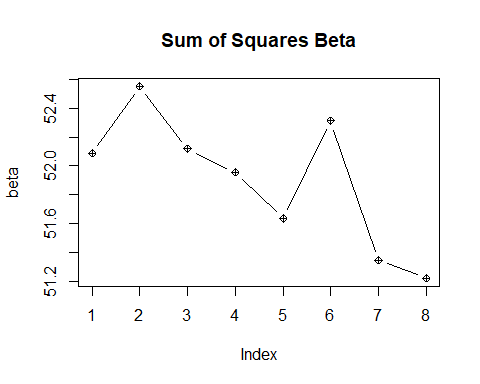
我們可以發現我們設定的所有變數都被Train Data 所配飾的Model篩選出來。

## The significant variables that we set : 4 7 8 10 13 14 17 19

## The Variabls select in the best model using train data: (Intercept) X4 X7 X8 X10 X13 X14 X17 X19

#### (g)

此題計算，  
計算在不同Model Size下的距離，由圖可知當變數個數為8時，有最小的Distance，  
故為表現最佳的模型，但此方法並不如其他方法來的穩健， 可以看到Distance先隨著變數個數下降，但中間又陡升，最後變數個數為8時，才降至最低。



附錄：(Code)

library(tidyverse)

library(ISLR)

library(leaps)

### (8)

#### (a)

set.seed(1234567)

X = rnorm(100)

e = rnorm(100)

par(mfrow=c(1,2))

plot(X,main="X")

plot(e,main="e")

#### (b)

#### (c)

predictors = sapply(1:10, function(a){

X^a

})

data = cbind(predictors,Y) %>% as.data.frame()

model = regsubsets(x=data[,1:10],y=data[,11],data=data)

summary(model)

modelsum = summary(model)

par(mfrow=c(1,3))

modelsum$cp %>% plot(lwd =1.7, cex = .8,pch= 5,type="b",main= "Selected by Cp")

paste("Cp Select:",which.min(modelsum$cp),"Variables")

coef(model,which.min(modelsum$cp))

modelsum$bic %>% plot(lwd =1.7, cex = .8,pch= 8,type="b",main= "Selected by BIC")

paste("BIC Select:",which.min(modelsum$bic),"Variables")

coef(model,which.min(modelsum$bic))

# Adj R^2

modelsum$adjr2 %>% plot(lwd =1.7, cex = .8,pch= 7,type="b",main= "Selected by adjusted R^2")

paste("Adj R^2 Select:",which.max(modelsum$adjr2),"Variables")

coef(model,which.max(modelsum$adjr2))

#### (d)

model\_f = regsubsets(x=data[,1:10],y=data[,11],data=data,method = "forward")

modelf\_sum = summary(model\_f)

par(mfrow=c(1,3))

modelf\_sum$cp %>% plot(lwd =1.7, cex = .8,pch= 5,type="b",main= "Selected by Cp")

paste("Cp Select:",which.min(modelf\_sum$cp),"Variables")

coef(model,which.min(modelf\_sum$cp))

modelf\_sum$bic %>% plot(lwd =1.7, cex = .8,pch= 8,type="b",main= "Selected by BIC")

paste("BIC Select:",which.min(modelf\_sum$bic),"Variables")

coef(model,which.min(modelf\_sum$bic))

modelf\_sum$adjr2 %>% plot(lwd =1.7, cex = .8,pch= 7,type="b",main= "Selected by adjusted R^2")

paste("Adj R^2:",which.max(modelf\_sum$adjr2),"Variables")

coef(model,which.max(modelf\_sum$adjr2))

model\_f = regsubsets(x=data[,1:10],y=data[,11],data=data,method = "backward")

modelf\_sum = summary(model\_f)

par(mfrow=c(1,3))

# Cp

modelf\_sum$cp %>% plot(lwd =1.7, cex = .8,pch= 5,type="b",main= "Selected by Cp")

paste("Cp Select:",which.min(modelf\_sum$cp),"Variables")

coef(model,which.min(modelf\_sum$cp))

# BIC

modelf\_sum$bic %>% plot(lwd =1.7, cex = .8,pch= 8,type="b",main= "Selected by BIC")

paste("BIC Select:",which.min(modelf\_sum$bic),"Variables")

coef(model,which.min(modelf\_sum$bic))

# Adj R^2

modelf\_sum$adjr2 %>% plot(lwd =1.7, cex = .8,pch= 7,type="b",main= "Selected by adjusted R^2")

paste("Adj R^2:",which.max(modelf\_sum$adjr2),"Variables")

coef(model,which.max(modelf\_sum$adjr2))

#### (e)

require(glmnet)

d = as.matrix(data)

CV\_lasso = cv.glmnet(x = d[,-11],y = d[,11],

family = "gaussian",nfold = 10,alpha = 1)

plot(CV\_lasso)

CV\_lasso

lasso\_min = glmnet(x = d[,-11],y = d[,11],

family = "gaussian",alpha = 1,lambda = CV\_lasso$lambda.min)

lasso\_lse = glmnet(x = d[,-11],y = d[,11],

family = "gaussian",alpha = 1,lambda = CV\_lasso$lambda.1se)

print(lasso\_min$beta)

print(lasso\_lse$beta)

#### (f)

Y\_1 = 5+7\*X^7+e

data = cbind(predictors,Y\_1) %>% as.data.frame()

model = regsubsets(Y\_1~.,data=data)

modelsum = summary(model)

par(mfrow=c(1,3))

# Cp

modelsum$cp %>% plot(lwd =1.7, cex = .8,pch= 5,type="b",main= "Selected by Cp")

paste("Cp Select:",which.min(modelf\_sum$cp),"Variables")

coef(model,which.min(modelsum$cp))

# BIC

modelsum$bic %>% plot(lwd =1.7, cex = .8,pch= 8,type="b",main= "Selected by BIC")

paste("BIC Select:",which.min(modelf\_sum$bic),"Variables")

coef(model,which.min(modelsum$bic))

# Adj R^2

modelsum$adjr2 %>% plot(lwd =1.7, cex = .8,pch= 7,type="b",main= "Selected by adjusted R^2")

paste("Adj R^2:",which.max(modelf\_sum$adjr2),"Variables")

coef(model,which.max(modelsum$adjr2))

require(glmnet)

d = as.matrix(data)

CV\_lasso = cv.glmnet(x = d[,-11],y = d[,11],

family = "gaussian",nfold = 10,alpha = 1)

plot(CV\_lasso)

CV\_lasso

lasso\_min = glmnet(x = d[,-11],y = d[,11],

family = "gaussian",alpha = 1,lambda = CV\_lasso$lambda.min)

lasso\_lse = glmnet(x = d[,-11],y = d[,11],

family = "gaussian",alpha = 1,lambda = CV\_lasso$lambda.1se)

print(lasso\_min$beta)

print(lasso\_lse$beta)

### (10)

#### (a)

set.seed(1234567)

X = matrix(rnorm(20000),ncol=20)

b = c(0.001,0.0001,0.001,13,0.004,

0.007,15,16,0.005,17,

0.008,-0.002,19,20,-0.001,

0.005,22,0.003,23,0.003) %>% as.matrix(ncol=1)

e = rnorm(1000)

Y = X%\*%b+e

cat("X",str(X))

cat("beta",str(b))

cat("Y",str(Y))

data = cbind(X,Y)

#### (b)

set.seed(1122)

index = sample(1:1000,100,replace = F)

data = as.data.frame(data)

names(data) = c(paste0("X",seq(1:20)),"Y")

train = data[index,]

test = data[-index,]

cat("Train:",dim(train)); cat("Test:",dim(test))

#### (c)

model = regsubsets(x = train[,1:20],y = train[,21],data=train)

model\_sum = summary(model)

MSE = model\_sum$rss/20

plot(MSE,lwd =1.7, cex = .8,pch= 5,type="b",main= "Selected by MSE")

#### (d)

ind = model\_sum$which %>% .[1,-1] #去掉截距項

sub = train[,ind] %>% cbind(Y = train$Y) %>% as.data.frame()

names(sub) = c("X19","Y")

model = lm(Y~.,data=sub)

Y\_hat = predict(model,newdata = test[,-21])

M\_1=sum((Y\_hat-test$Y)^2)/dim(sub)[2]-1 #sub扣掉Y其他為解釋變數

MSE = sapply(2:8, function(a){

ind = model\_sum$which %>% .[a,-1] #去掉截距項

sub = train[,ind] %>% cbind(Y = train$Y)

model = lm(Y~.,data=sub)

Y\_hat = predict(model,newdata = test[,-21])

sum((Y\_hat-test$Y)^2)/dim(sub)[2]-1 #sub扣掉Y其他為解釋變數

})

MSE = c(M\_1,MSE)

plot(MSE,lwd =1.7, cex = .8,pch= 8,type="b",main= "MSE of Test by each Size")

#### (e)

ind = model\_sum$which %>% .[8,-1] #去掉截距項

sub = train[,ind] %>% cbind(Y = train$Y) %>% as.data.frame()

model\_sum$which %>% .[which.min(MSE),] %>% as.matrix()

model = lm(Y~.,data=sub)

Y\_hat = predict(model,newdata = test[,-21])

M\_1=sum((Y\_hat-test$Y)^2)/dim(sub)[2]-1 #sub扣掉Y其他為解釋變數

paste("MSE of the Best Model in All Model Size(8 Variables): ",M\_1)

#### (f)

我們可以發現我們設定的所有變數都被Train Data 所配飾的Model篩選出來。

cat("The significant variables that we set : ",which(b>1))

cat("The Variabls select in the best model using train data: ",model\_sum$which %>% .[which.min(MSE),] %>% which(TRUE) %>% names())

model\_full = lm(Y~.,data=data)

model\_sub = regsubsets(Y~.,data=data)

model\_sub\_sum = summary(model\_sub)

coef = matrix(0,ncol = 20,nrow = 8) %>% as.data.frame()

names(coef) = c(paste0("X",seq(1:20)))

for (a in 2:8) {

ind = model\_sub\_sum$which %>% .[a,-1] #去掉截距項

sub = data[,ind] %>% cbind(Y = train$Y) %>% as.data.frame()

lm\_sub = lm(Y~.,data = sub)

coef[a,which(colnames(data) %in% names(lm\_sub$coefficients)[-1])] = lm\_sub$coefficients[-1]

}

beta = sapply(1:8, function(a){

sqrt(sum((model\_full$coefficients[-1] - coef[a,])^2))

})

plot(beta,type = "b",pch= 10,main = "Sum of Squares Beta")