Statistical Learning Homework 3

110024516 邱繼賢

Problem 1.

```
library(AppliedPredictiveModeling) #install package first!!
library(corrplot)  #correlation plot
library(leaps)
library(latex2exp)
library(glmnet)
library(knitr)
library(pls)
```

```
data(ChemicalManufacturingProcess)
dim(ChemicalManufacturingProcess)
```

```
## [1] 176  58
```

- 原始資料共有 176 筆觀測值，58 個變數，其中有 12 個和 biological starting material 有關，45 個和 manufacturing process 有關，剩餘 1 個為 response variable *Yield*

```
sum(is.na(ChemicalManufacturingProcess))
```

```
## [1] 106
```

- 發現資料中共有 106 個數值缺失，將有缺失值的 observation 移除

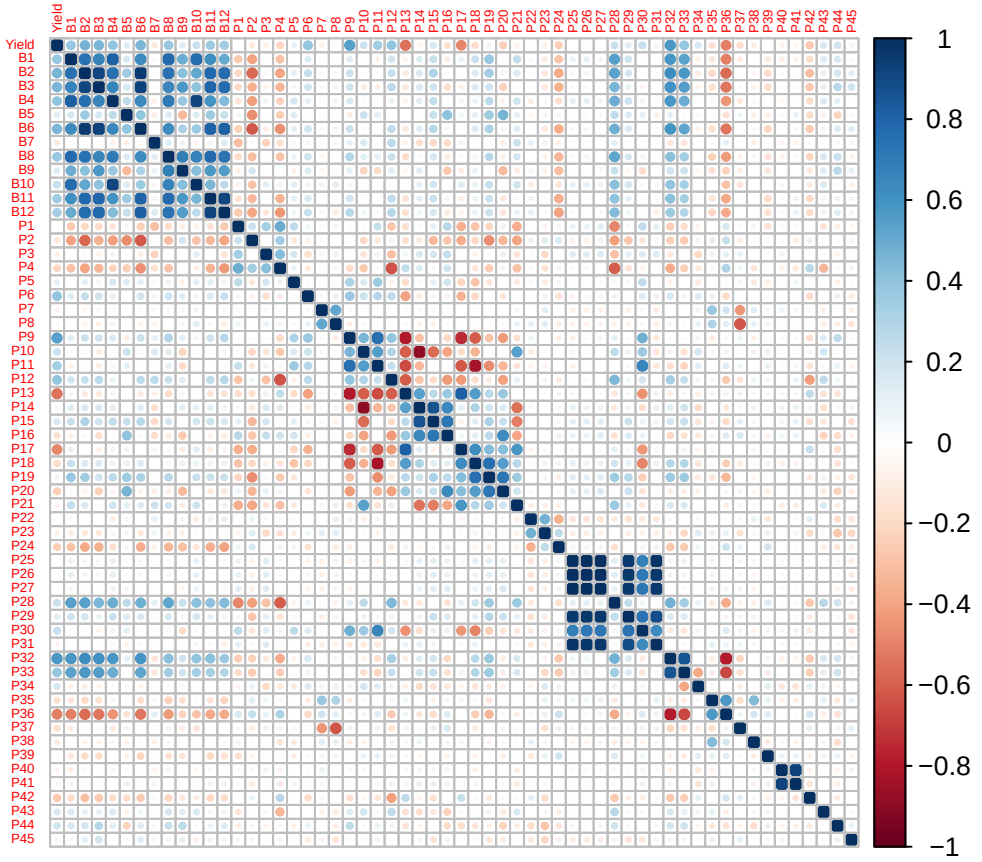
```
CMP <- na.omit(ChemicalManufacturingProcess) #remove missing data
dim(CMP)
```

```
## [1] 152  58
```

- 剩餘資料僅剩 152 筆觀測值，但都沒有任何數值有所缺失，以下分析就使用此資料

```
#rename variable
B_name = c()
for (i in 1:12){
  B_name[i] = paste("B",i,sep="")
}
P_name = c()
for (i in 1:45){
  P_name[i] = paste("P",i,sep="")
}
names(CMP) <- c("Yield",B_name, P_name)

corrplot::corrplot(cor(CMP), tl.cex = 0.4)
```



- biological starting material 的變數間大多呈現中至高度正相關，建構模型時可能會具有共線性

- manufacturing process 的變數間有正相關也有負相關，*P25~P31* 間互相有著高度正相關

將資料以 120:32 的比例隨機分割成 train & test data set，以下各種建構模型方法皆是對 train data 建模，然後在 test data 上比較其表現

```
set.seed(1116)
idx = sample(1:152,120)
train_data = CMP[idx,]
test_data = CMP[-idx,]
```

Subset Selection via Criterion based

利用 Cp, BIC, R_a^2 等 criterion 進行 model selection，以此來決定模型中應該保留的變數個數。因為全部共有 57 個解釋變數，總共有 2^{57} 種模型選擇，若全部模型都對 criterion 計算會太花時間，故此僅使用 forward 的方式選取模型：

```
regfit = regsubsets(Yield~., train_data, nvmax=57,
                    really.big = T,method = "forward")
```

```
## Reordering variables and trying again:
```

```
regfit_sum = summary(regfit)
```

```
which.min(regfit_sum$cp)
```

```
## [1] 13
```

```
which.min(regfit_sum$bic)
```

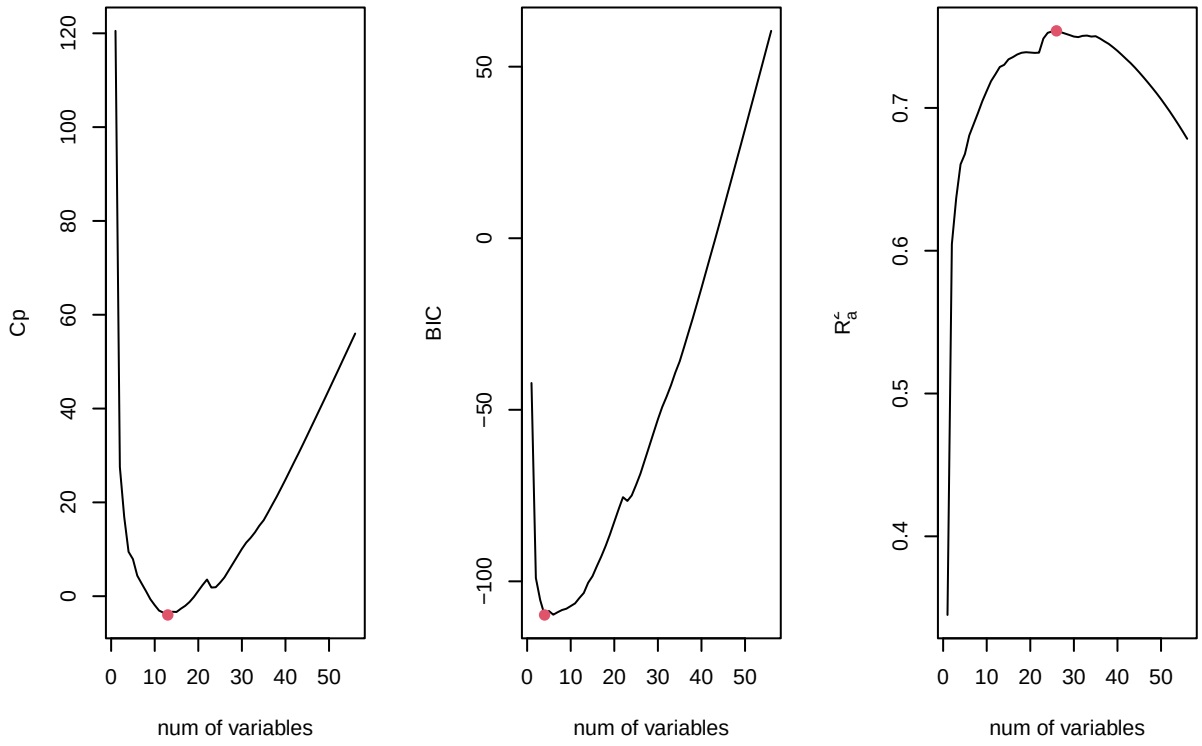
```
## [1] 4
```

```
which.max(regfit_sum$adjr2)
```

```
## [1] 26
```

```
par(mfrow = c(1,3))
plot(regfit_sum$cp,type="l",ylab="Cp",xlab="num of variables")
points(13,regfit_sum$cp[13],col=2,pch=16,cex=1.2)
plot(regfit_sum$bic,type="l",ylab="BIC",xlab="num of variables")
```

```
points(4,regfit_sum$bic[4],col=2,pch=16,cex=1.2)
plot(regfit_sum$adjr2,type="l",ylab=TeX("$R^2_a$"),xlab="num of variables")
points(26,regfit_sum$adjr2[26],col=2,pch=16,cex=1.2)
```



- 三種 criterion 所決定的解釋變數個數分別為：13, 4, 26

所選取出變數係數的估計值如下：

```
round(coef(regfit,13),3)
```

```
## (Intercept)          B5          P1          P7          P9         P19
##     -38.570       0.179       0.452      -0.267       0.696       0.017
##         P20         P23         P25         P29         P33         P34
##      -0.013      -0.031       0.000       0.068       0.287      10.459
##         P38         P40
##      -0.042      -1.487
```

```
round(coef(regfit,4),3)
```

```
## (Intercept)          P1          P9         P33         P34
##     -52.885       0.306       0.692       0.429      12.395
```

```
round(coef(regfit,26),3)
```

```
## (Intercept)          B5          B7         B11          P1          P2
##     291.901       0.265      -2.286      -0.042       0.232      -0.019
##          P3          P4          P7          P9         P11         P12
##      -3.671       0.073      -0.477       0.874      -1.280       0.000
##         P19         P20         P23         P25         P28         P29
##       0.000      -0.012      -0.125      -0.006      -0.066       1.654
##         P30         P32         P33         P34         P37         P38
##      -0.175       0.249      -0.327       1.921      -0.756      -0.043
##         P40         P44         P21
##      -0.490      -0.393      -0.155
```

Subset Selection via Cross-Validation

將 train data 隨機分割成五份做 5-fold CV，然後分別計算模型在不同解釋變數個數下對 Validation set 的 MSE，以此來決定模型解釋變數個數：

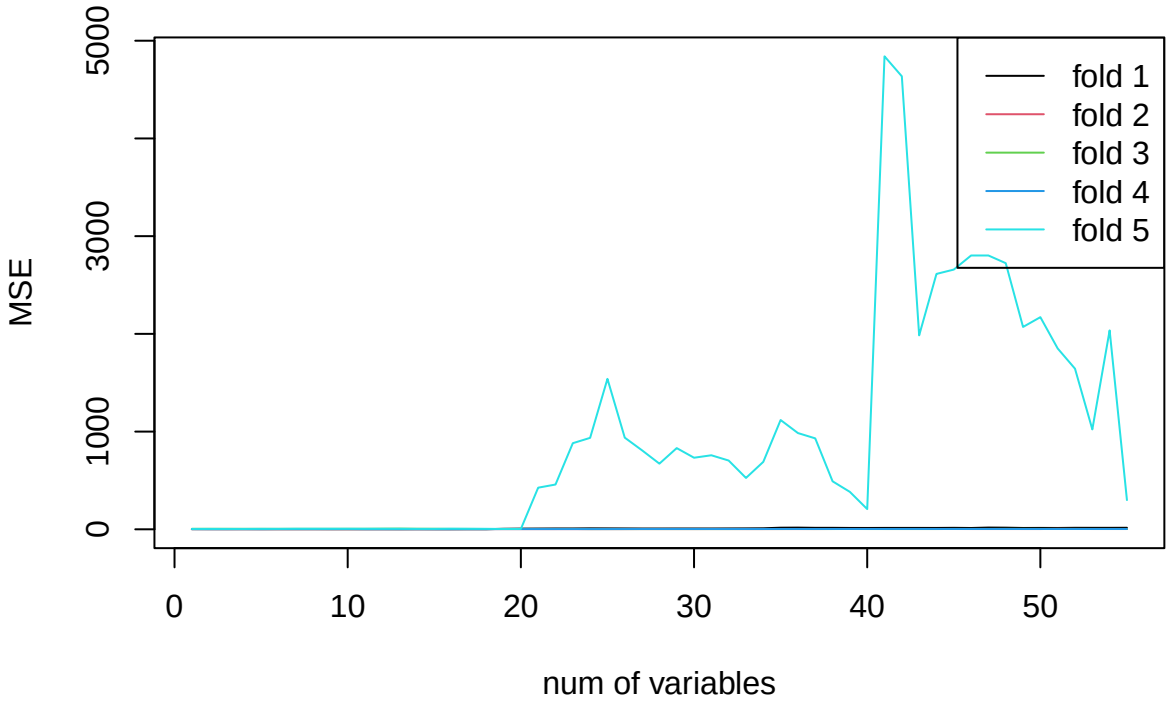
```
k = 5 ; n = dim(train_data)[1] ; p = dim(train_data)[2]-3
set.seed(11137)
fold_idx = sample(rep(1:k, length = n))
cv.errors = matrix(NA, k, p)
```

```
predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars] %*% coefi
 }
```

```
for (j in 1:k) {
    best.fit = regsubsets(Yield~., data=train_data[fold_idx!=j,],
                          nvmax = p,really.big = T,method = "backward")
    for (i in 1:p) {
        pred = predict.regsubsets(best.fit, train_data[fold_idx==j,],id=i)
        cv.errors[j,i] = mean((train_data$Yield[fold_idx==j]-pred)^2)
    }
}
```

```
## Reordering variables and trying again:
## Reordering variables and trying again:
## Reordering variables and trying again:
## Reordering variables and trying again:
## Reordering variables and trying again:
```

```
plot(cv.errors[1,], type="l", ylim = c(min(cv.errors),max(cv.errors)),
     ylab="MSE", xlab="num of variables")
for (i in 2:5) {
    points(1:p,cv.errors[i,],type="l",col=i)
}
legend("topright",legend = paste("fold",1:5),lty=1,col=1:5)
```

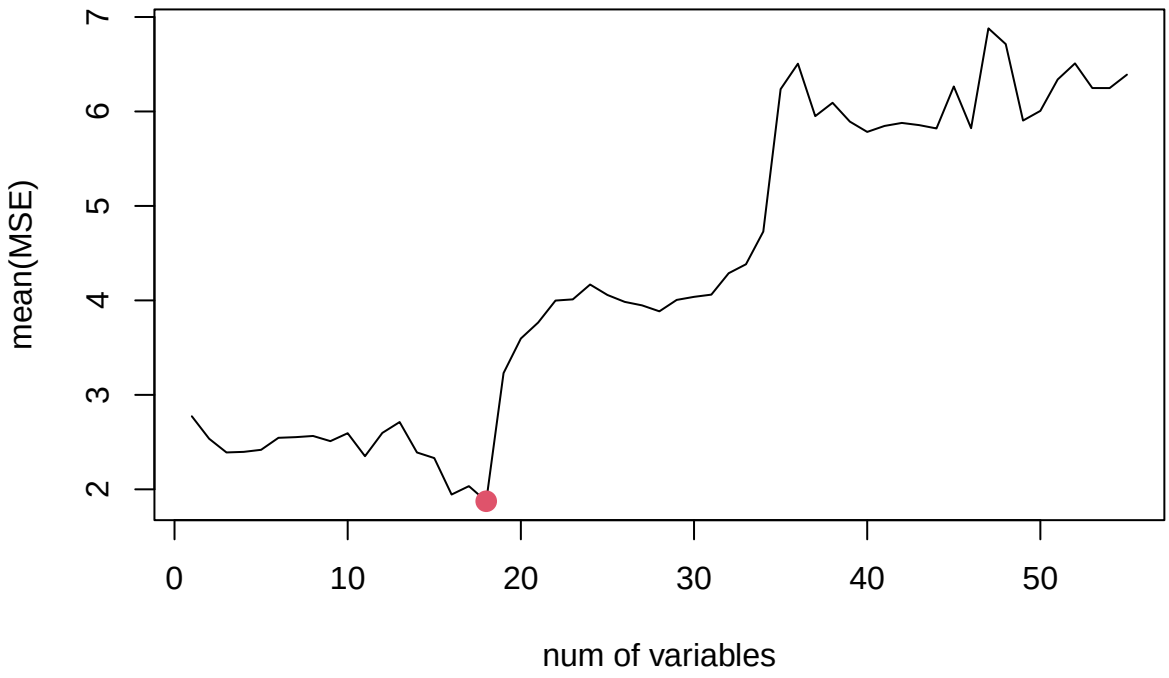


- fold 5 時計算出的 MSE 非常大，可能因為此時的 Validation set 被分割到了一些 outliers，故我們不考慮此情況，僅將剩餘四種 fold 所計算的 MSE 平均

```
cv.errors_mean = apply(cv.errors[-5,],2,mean) ; which.min(cv.errors_mean)
```

```
## [1] 18
```

```
plot(cv.errors_mean,type="l", ylab="mean(MSE)", xlab="num of variables")
points(18,cv.errors_mean[18],pch=16,col=2,cex=1.5)
```



- 在選取 18 個解釋變數時，平均的 MSE 最小

將所有資料合併，選取 18 個解釋變數的模型估計係數如下：

```
best.fit_full = regsubsets(Yield~., train_data, nvmax=p,
                           really.big = T, method = "forward")
```

```
## Reordering variables and trying again:
```

```
round(coef(best.fit_full,18),3)
```

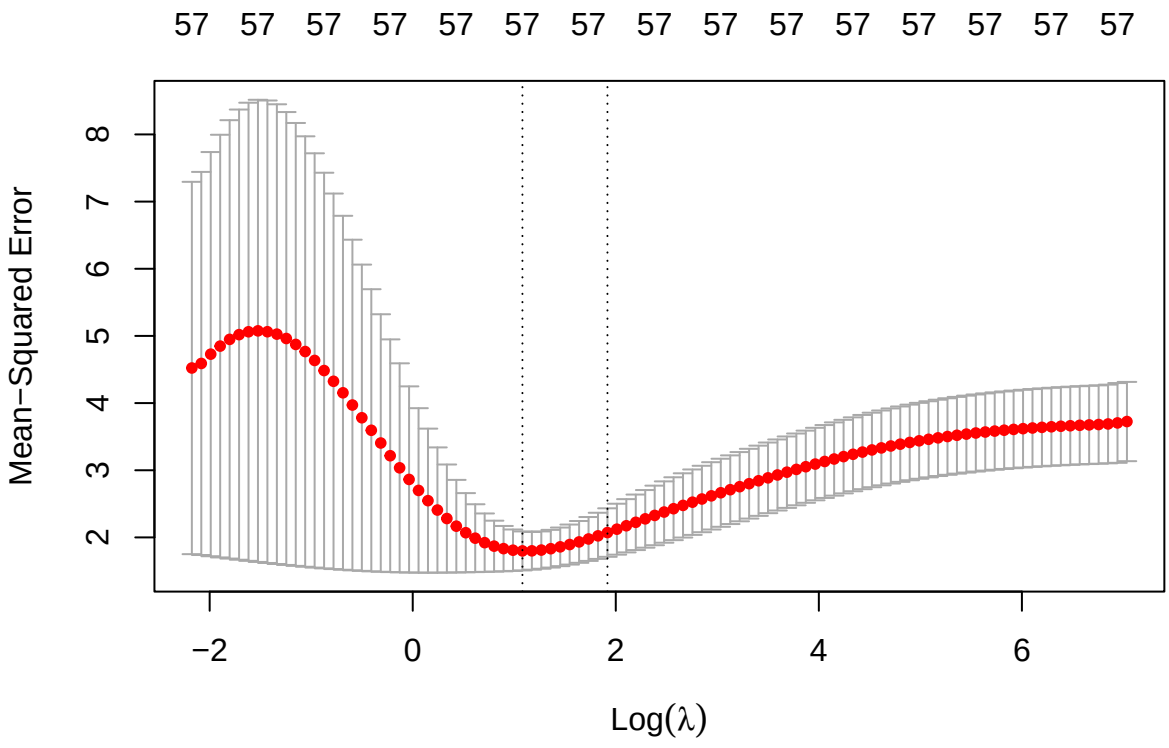
```
## (Intercept)          B5          B7          P1          P2          P4
##     316.524       0.255      -3.217       0.329       0.002       0.106
##          P7          P9         P11         P12         P19         P20
##      -0.351       0.903      -1.729       0.000      -0.003      -0.011
##         P23         P25         P29         P33         P34         P38
##      -0.019      -0.008       1.970       0.208      11.550       0.112
##         P40
##      -0.056
```

```
pred.cp = predict.regsubsets(best.fit_full,test_data,id=13)
MSE.cp = mean((test_data$Yield-pred.cp)^2)
pred.bic = predict.regsubsets(best.fit_full, test_data, id=4)
MSE.bic = mean((test_data$Yield-pred.bic)^2)
pred.adjr2 = predict.regsubsets(best.fit_full,test_data,id=26)
MSE.adjr2 = mean((test_data$Yield-pred.adjr2)^2)
pred.5cv = predict.regsubsets(best.fit_full,test_data,id=18)
MSE.5cv = mean((test_data$Yield-pred.5cv)^2)
```

Ridge Regression via Cross-Validation

利用 5-fold CV 計算 MSE 的平均，以選取 Ridge Regression 所使用的參數 λ

```
set.seed(1114)
x = as.matrix(train_data[,-1]) ; y = train_data$Yield
cv.ridge = cv.glmnet(x,y, alpha=0, nfolds = 5)
plot(cv.ridge)
```



mean(MSE) 最小時的 λ 為：

```
best.lam_ridge = cv.ridge$lambda.min
best.lam_ridge
```

```
## [1] 2.946562
```

將全部資料合併用以配飾 Ridge Regression，且帶入前面所求得的 λ，然而因為 Ridge Regression 並沒有辦法使得變數的係數真的為零，以達到 model selection 的目的，故以下僅列出 $|\hat{\beta}_i| > 0.05$ 的那些係數視為重要的解釋變數：

```
full.ridge = glmnet(x,y,alpha=0)
coef.ridge = predict(full.ridge, type="coefficients"
                     ,s=best.lam_ridge)[1:58,]
round(coef.ridge[abs(coef.ridge)>0.05],3)
```

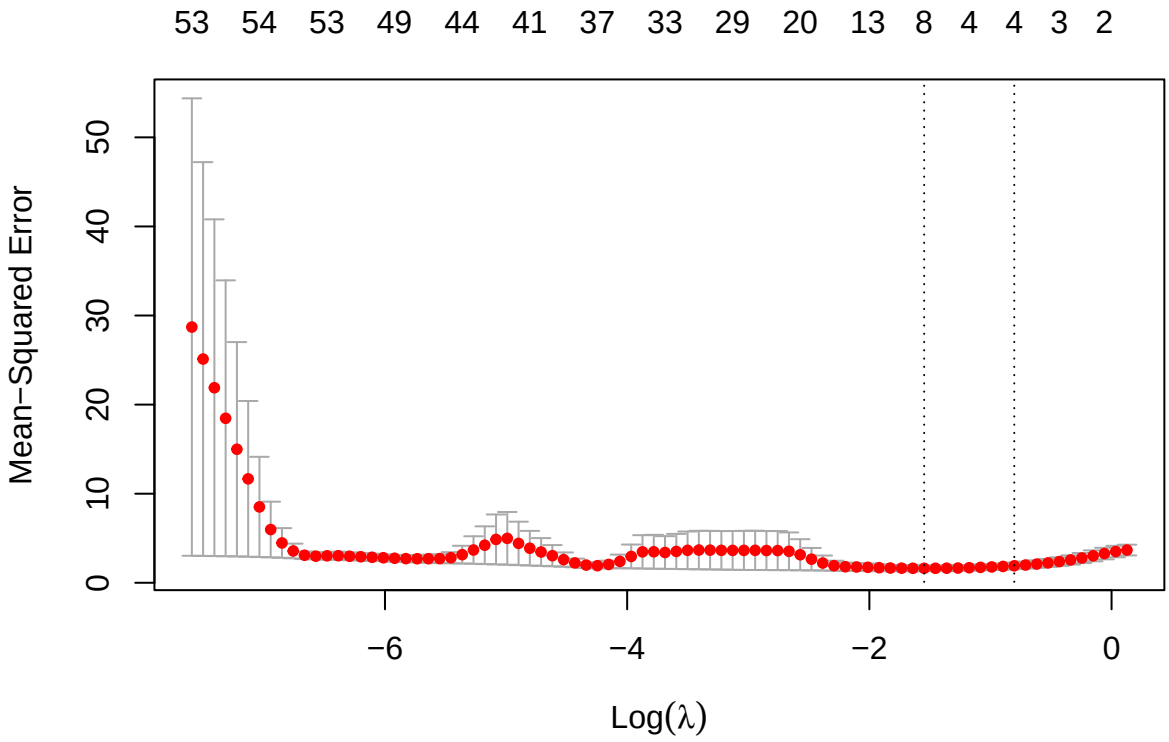
```
## (Intercept)          B1          B7          B8          P3          P7
##      87.497       0.091      -0.640       0.065      -0.893      -0.130
##          P9         P11         P13         P17         P21         P34
##       0.125       0.138      -0.164      -0.139      -0.126       2.284
##         P36         P37         P40         P41         P42         P43
##    -186.492      -0.257      -1.099      -1.089      -0.129       0.052
##         P44         P45
##       0.168       0.067
```

```
x_test = as.matrix(test_data[,-1])
pred.ridge = predict(full.ridge, newx = x_test,s=best.lam_ridge)
MSE.ridge = mean((test_data$Yield-pred.ridge)^2)
```

Lasso Regression via Cross-Validation

利用 5-fold CV 計算 MSE 的平均，以選取 Lasso Regression 所使用的參數 λ

```
set.seed(1114)
cv.lasso = cv.glmnet(x,y,alpha = 1, nfolds=5)
plot(cv.lasso)
```



mean(MSE) 最小時的 λ 為：

```
best.lam_lasso = cv.lasso$lambda.min
best.lam_lasso
```

```
## [1] 0.2127656
```

將全部資料合併用以配飾 Lasso Regression，且帶入前面所求得的 λ，和 Ridge 不同的是，Lasso 可以使得變數的係數真的為零，以達到 model selection 的效果，故以下列出係數不為零的變數視為重要解釋變數：

```
full.lasso = glmnet(x,y,alpha = 1)
coef.lasso = predict(full.lasso, type="coefficients",
                     s=best.lam_lasso)[1:58,]
round(coef.lasso[coef.lasso!=0],3)
```

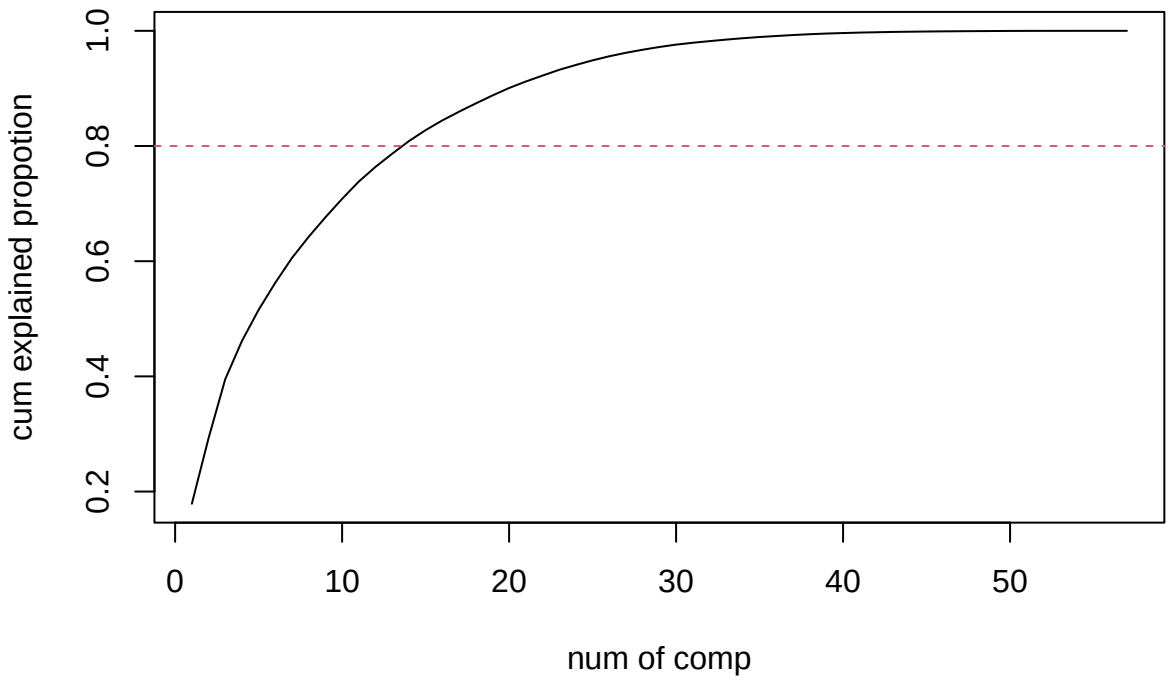
```
## (Intercept)          B6          P6          P9         P17         P32
##      15.167       0.008       0.004       0.362      -0.283       0.121
##         P34         P36         P37
##       0.859    -208.990      -0.058
```

```
pred.lasso = predict(full.lasso, newx = x_test,s=best.lam_lasso)
MSE.lasso = mean((test_data$Yield-pred.lasso)^2)
```

Principal Components Regression

對 train data 做 PCR，並利用累積解釋比例超過 80% 來選取 component 個數

```
fit.pca = pcr(Yield~., data=train_data, scale=T)
plot(cumsum(fit.pca$Xvar)/fit.pca$Xtotvar,type="l",
     xlab = "num of comp", ylab = "cum explained propotion")
abline(h=0.8,lty=2,col=2)
```



⇒ 選出 15 個 components 來做為模型的解釋變數，係數如下所示：

```
round(fit.pca$coefficients[,,1:15],2)
```

```
##      1 comps 2 comps 3 comps 4 comps 5 comps 6 comps 7 comps 8 comps 9 comps
## B1      0.08    0.07    0.07    0.07    0.07    0.06    0.05    0.06    0.06
```

```
## B2     0.09    0.09    0.09    0.10    0.10    0.10    0.09    0.09    0.09
## B3     0.08    0.09    0.09    0.10    0.11    0.11    0.11    0.09    0.08
## B4     0.07    0.07    0.07    0.08    0.08    0.08    0.07    0.04    0.04
## B5     0.03    0.00    0.00    0.02    0.01    0.01    0.00    0.04    0.05
## B6     0.08    0.08    0.08    0.09    0.09    0.09    0.09    0.09    0.09
## B7     0.00    0.01    0.01   -0.02   -0.02   -0.01   -0.01    0.02   -0.07
## B8     0.08    0.09    0.09    0.10    0.10    0.10    0.09    0.09    0.03
## B9     0.05    0.07    0.07    0.07    0.08    0.08    0.07    0.05   -0.05
## B10    0.05    0.04    0.05    0.05    0.05    0.04    0.03    0.00   -0.02
## B11    0.07    0.08    0.08    0.10    0.10    0.10    0.10    0.10    0.02
## B12    0.07    0.09    0.08    0.09    0.09    0.10    0.10    0.09    0.02
## P1    -0.03   -0.02    0.00    0.04    0.04    0.04    0.02    0.05    0.05
## P2    -0.04    0.00    0.00    0.00    0.01    0.01    0.02    0.04    0.01
## P3    -0.01   -0.02   -0.01    0.01    0.01    0.01   -0.01   -0.05   -0.04
## P4    -0.05   -0.06   -0.06   -0.03   -0.02   -0.02   -0.04   -0.05   -0.07
## P5     0.00    0.02    0.04    0.07    0.07    0.06    0.05    0.02    0.02
## P6     0.02    0.06    0.07    0.08    0.08    0.08    0.09    0.14    0.13
## P7     0.00    0.01    0.00    0.01    0.00   -0.02   -0.01   -0.01   -0.02
## P8     0.00    0.01    0.00    0.00    0.00   -0.03   -0.03    0.00    0.00
## P9     0.02    0.11    0.13    0.15    0.15    0.15    0.15    0.17    0.18
## P10    0.00    0.06    0.07    0.02    0.02    0.02    0.01    0.02    0.01
## P11    0.00    0.08    0.11    0.13    0.12    0.12    0.11    0.11    0.14
## P12    0.04    0.10    0.10    0.08    0.08    0.08    0.10    0.11    0.15
## P13   -0.01   -0.11   -0.13   -0.13   -0.13   -0.13   -0.13   -0.16   -0.19
## P14    0.02   -0.05   -0.05    0.01    0.01    0.01    0.02    0.01    0.03
## P15    0.03   -0.02   -0.02    0.05    0.05    0.05    0.06    0.07    0.11
## P16   -0.01   -0.07   -0.07   -0.01   -0.01   -0.01   -0.01    0.00    0.02
## P17    0.00   -0.09   -0.12   -0.15   -0.15   -0.15   -0.16   -0.19   -0.21
## P18    0.02   -0.07   -0.09   -0.11   -0.11   -0.11   -0.11   -0.08   -0.09
## P19    0.03   -0.05   -0.06   -0.07   -0.07   -0.07   -0.07   -0.03   -0.03
## P20   -0.01   -0.09   -0.10   -0.09   -0.10   -0.10   -0.10   -0.06   -0.06
## P21    0.01   -0.01   -0.02   -0.09   -0.09   -0.10   -0.11   -0.11   -0.11
## P22    0.00    0.02    0.01    0.03    0.03    0.03    0.03    0.12    0.04
## P23   -0.01    0.00    0.00    0.01    0.01    0.00    0.03    0.09    0.01
## P24   -0.04   -0.05   -0.04   -0.04   -0.04   -0.04   -0.03   -0.08   -0.06
## P25    0.01   -0.04    0.01   -0.01   -0.01   -0.01    0.00    0.00   -0.01
```

```
## P26    0.01   -0.04    0.01    0.00    0.00    0.00    0.00    0.01    0.00
## P27    0.01   -0.05    0.01   -0.01   -0.01   -0.01   -0.01    0.01   -0.01
## P28    0.06    0.06    0.06    0.03    0.02    0.03    0.03    0.02    0.04
## P29    0.02   -0.03    0.02    0.00    0.00    0.00    0.01    0.02    0.02
## P30    0.00    0.02    0.07    0.07    0.07    0.07    0.07    0.08    0.09
## P31    0.00   -0.05    0.00   -0.01   -0.01   -0.01   -0.01   -0.01   -0.03
## P32    0.07    0.07    0.06    0.05    0.06    0.06    0.06    0.09    0.18
## P33    0.06    0.05    0.04    0.03    0.04    0.03    0.04    0.03    0.13
## P34   -0.01    0.01    0.01    0.01    0.01    0.01    0.01    0.07    0.03
## P35   -0.02   -0.03   -0.03   -0.03   -0.02   -0.05   -0.05   -0.06    0.00
## P36   -0.06   -0.06   -0.05   -0.05   -0.06   -0.07   -0.07   -0.10   -0.15
## P37   -0.01   -0.03   -0.02   -0.02   -0.02    0.01    0.00   -0.01   -0.02
## P38    0.00   -0.01   -0.01   -0.01   -0.01   -0.03   -0.02   -0.01    0.05
## P39   -0.01    0.00    0.00   -0.02   -0.02   -0.02   -0.01   -0.06   -0.04
## P40    0.00    0.00    0.01    0.01    0.01    0.01    0.02   -0.13   -0.19
## P41    0.01    0.01    0.02    0.02    0.02    0.01    0.03   -0.12   -0.18
## P42   -0.03   -0.05   -0.05   -0.06   -0.06   -0.06   -0.08   -0.06   -0.12
## P43    0.02    0.04    0.04    0.01    0.01    0.01    0.01    0.00    0.02
## P44    0.02    0.06    0.06    0.05    0.06    0.06    0.06    0.03    0.07
## P45    0.00    0.01    0.00    0.00    0.00    0.00    0.00   -0.05    0.00
##     10 comps 11 comps 12 comps 13 comps 14 comps 15 comps
## B1      0.05     0.07     0.06     0.06     0.06     0.05
## B2      0.10     0.09     0.09     0.09     0.10     0.08
## B3      0.08     0.06     0.07     0.07     0.08     0.07
## B4      0.03     0.04     0.05     0.06     0.05     0.03
## B5      0.05     0.03     0.06     0.06     0.05     0.02
## B6      0.10     0.07     0.08     0.08     0.09     0.07
## B7     -0.07    -0.08    -0.10    -0.10    -0.10    -0.24
## B8      0.02     0.02     0.03     0.02     0.01     0.01
## B9     -0.05    -0.06    -0.05    -0.06    -0.07    -0.04
## B10    -0.04    -0.01    -0.01     0.00    -0.01    -0.02
## B11     0.02    -0.01     0.00     0.00     0.00    -0.02
## B12     0.02    -0.01    -0.01    -0.01    -0.01     0.00
## P1      0.05     0.07     0.09     0.10     0.08     0.11
## P2      0.01     0.01     0.03     0.02     0.00     0.01
## P3     -0.04    -0.03    -0.05    -0.05    -0.06    -0.01
```

```
## P4    -0.07    -0.03    -0.03    -0.04    -0.03    -0.04
## P5     0.02     0.03     0.02     0.04     0.03    -0.02
## P6     0.13     0.15     0.16     0.15     0.16     0.25
## P7    -0.02    -0.02    -0.02    -0.03    -0.02     0.00
## P8     0.01     0.03     0.04     0.04     0.05     0.05
## P9     0.17     0.17     0.15     0.14     0.15     0.16
## P10    0.02     0.02     0.02     0.02     0.02     0.03
## P11    0.14     0.14     0.13     0.13     0.14     0.12
## P12    0.15     0.15     0.14     0.15     0.13     0.11
## P13   -0.19    -0.20    -0.20    -0.19    -0.18    -0.20
## P14    0.03     0.03     0.02     0.02     0.02     0.02
## P15    0.10     0.10     0.09     0.09     0.09     0.11
## P16    0.02     0.01     0.00     0.00    -0.01     0.00
## P17   -0.21    -0.21    -0.22    -0.21    -0.20    -0.23
## P18   -0.09    -0.08    -0.08    -0.09    -0.10    -0.06
## P19   -0.03    -0.01    -0.01    -0.03    -0.04     0.02
## P20   -0.06    -0.06    -0.05    -0.07    -0.08    -0.04
## P21   -0.11    -0.11    -0.11    -0.11    -0.11    -0.13
## P22    0.04     0.04     0.00     0.01     0.01     0.03
## P23    0.00     0.01     0.01     0.02     0.01    -0.05
## P24   -0.08    -0.07    -0.04    -0.03    -0.04    -0.12
## P25   -0.01    -0.01    -0.01    -0.01    -0.01    -0.01
## P26    0.00     0.00     0.00     0.00     0.00     0.00
## P27   -0.01    -0.01    -0.01    -0.01    -0.01     0.00
## P28    0.03     0.03     0.04     0.03     0.03     0.02
## P29    0.02     0.02     0.02     0.02     0.02     0.02
## P30    0.09     0.08     0.08     0.08     0.09     0.08
## P31   -0.03    -0.03    -0.03    -0.03    -0.03    -0.03
## P32    0.18     0.20     0.20     0.22     0.22     0.24
## P33    0.14     0.17     0.15     0.16     0.16     0.13
## P34    0.02     0.00     0.04     0.06     0.07     0.18
## P35   -0.01    -0.07    -0.08    -0.09    -0.09    -0.11
## P36   -0.15    -0.20    -0.21    -0.23    -0.23    -0.27
## P37   -0.02    -0.06    -0.08    -0.07    -0.08    -0.06
## P38    0.04    -0.02    -0.03    -0.03    -0.03    -0.02
## P39   -0.06    -0.03    -0.03    -0.05    -0.04     0.01
```

```
## P40    -0.19    -0.17    -0.17    -0.17    -0.17    -0.11
## P41    -0.18    -0.16    -0.17    -0.16    -0.16    -0.12
## P42    -0.12    -0.15    -0.15    -0.13    -0.11    -0.05
## P43     0.01    -0.02    -0.04    -0.03    -0.05     0.12
## P44     0.07     0.07     0.08     0.04     0.06     0.05
## P45     0.01    -0.05    -0.03    -0.03    -0.03    -0.03
```

```
pred.pcr = predict(fit.pca, x_test, ncomp = 15)
MSE.pcr = mean((test_data$Yield-pred.pcr)^2)
```

Performance upon test data

各模型對 test data 的預測表現 MSE 計算結果呈現如下

```
table = data.frame(a = MSE.cp,b = MSE.bic,c = MSE.adjr2,
                   d = MSE.5cv,e = MSE.ridge,f = MSE.lasso,g = MSE.pcr)
rownames(table) = "MSE"
kable(table, col.names = c("Cp","BIC","$R^2_a$","5-fold CV","Ridge","Lasso","PCR"),
      digits = 3)
```

	Cp	BIC	R^2_a	5-fold CV	Ridge	Lasso	PCR
MSE	1.708	1.572	1.441	2.064	1.306	1.353	1.544

⇒ Ridge regression 在 test data 上的表現最好

Problem 2.

```
library(latex2exp)
library(boot)
data2 = read.csv("hw3_problem2.csv")
X = data2$x
n = dim(data2)[1]

compute_sigma.hat = function(X, idx) {
```

```
    X = X[idx]
    sd(X)
}
compute_sigma.tilde = function(X, idx) {
    X = X[idx]
    1.4826*median(abs(X-median(X)))
}
```

Compute estimated standard deviation for the whole data

$$\hat{\sigma} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(X_i - \bar{X}\right)^2} = 3.745385$$

$$\tilde{\sigma} = 1.4826 \times med_{1\leq i\leq n}\left\{|X_i - X_{med}|\right\} = 2.585234$$

```
sigma.hat = compute_sigma.hat(X, 1:n)
sigma.tilde = compute_sigma.tilde(X,1:n)
c(sigma.hat,sigma.tilde)
```

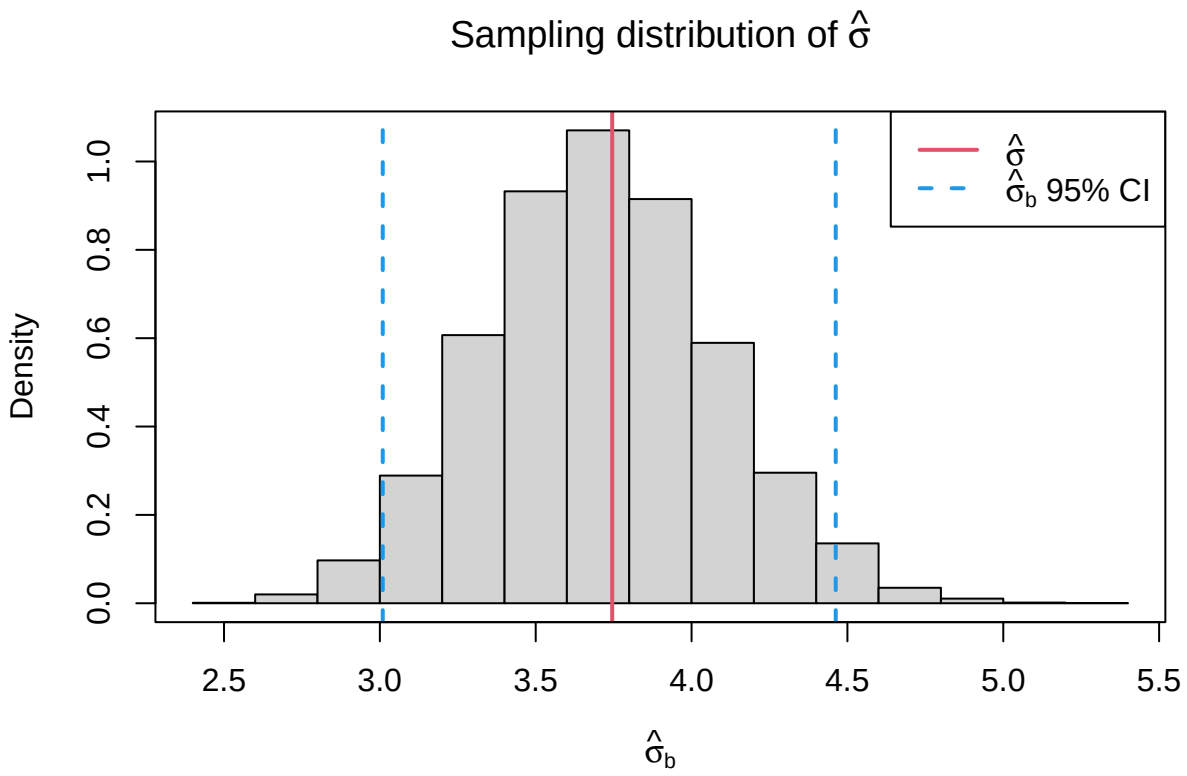
```
## [1] 3.745385 2.585234
```

And now apply bootstrap method (resample n observations with replacement from the raw data) 10000 times to compute $\hat{\sigma}_b^i$ and $\tilde{\sigma}_b^i$ for $i = 1, ..., 10000$

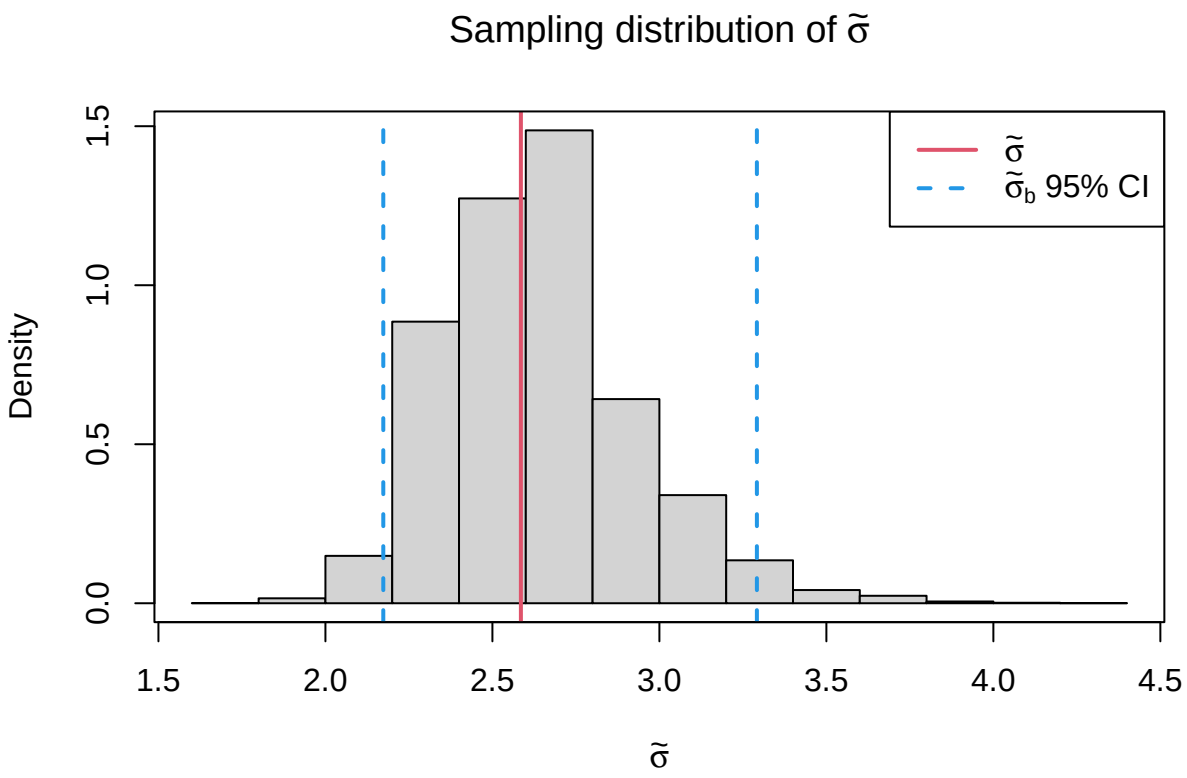
```
set.seed(1108)
sigma.hat_boots = boot(X,compute_sigma.hat,R=10000)
sigma.tilde_boots = boot(X,compute_sigma.tilde,R=10000)
```

Construct the sampling distribution of $\hat{\sigma}$ and $\tilde{\sigma}$ by histogram

```
hist(sigma.hat_boots$t[,1], probability = T,
     main = TeX("Sampling distribution of $\\hat{\\sigma}$"),
     xlab = TeX("$\\hat{\\sigma}_b$"))
abline(v = sigma.hat, col = 2, lwd = 2)
abline(v = quantile(sigma.hat_boots$t[,1],c(0.025,0.975)), col=4, lty=2, lwd=2)
legend("topright", legend = c(TeX("$\\hat{\\sigma}$"),TeX("$\\hat{\\sigma}_b$ 95% CI")),
       lty=c(1,2),col=c(2,4),lwd=2)
box()
```




```
hist(sigma.tilde_boots$t[,1], probability = T,
     main = TeX("Sampling distribution of $\\tilde{\\sigma}$"),
     xlab = TeX("$\\tilde{\\sigma}$"))
abline(v = sigma.tilde, col=2, lwd=2)
abline(v = quantile(sigma.tilde_boots$t[,1],c(0.025,0.975)),col=4,lwd=2,lty=2)
legend("topright", legend = c(TeX("$\\tilde{\\sigma}$"),TeX("$\\tilde{\\sigma}_b$ 95% CI")),
       lty=c(1,2),col=c(2,4),lwd=2)
box()
```



We can see that both $\hat{\sigma}$ and $\tilde{\sigma}$ are filled in the 95% bootstrap confidence intervals.

Compute the estimation of $var\left(\hat{\sigma}\right)$ and $var\left(\tilde{\sigma}\right)$

$$\hat{var}\left(\hat{\sigma}\right) = \frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\sigma}_b^i - \bar{\hat{\sigma}}_b\right)^2 = 0.13677879$$

$$\hat{var}\left(\tilde{\sigma}\right) = \frac{1}{B-1}\sum_{i=1}^{B}\left(\tilde{\sigma}_b^i - \bar{\tilde{\sigma}}_b\right)^2 = 0.08307997$$

```
c(var(sigma.hat_boots$t[,1]),var(sigma.tilde_boots$t[,1]))
```

```
## [1] 0.13677879 0.08307997
```