統計學習 作業三

106070038 科管院學士班 杜葳葳

1. 根據題目條件，建一個線性迴歸模型

yi =β1xi1+β2xi2+β3xi3+β4xi4+β5xi5+ε, i = 1,…,n

n = 100, σ = 5, β = (2, -2, 0.5, 1, -3), seed = 36

```
> cormat <- diag(1,nrow=5,ncol=5)
> cormat
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    0    0    0    0
[2,]    0    1    0    0    0
[3,]    0    0    1    0    0
[4,]    0    0    0    1    0
[5,]    0    0    0    0    1
```

```
> cormat[cormat==0] <- 0.5
> cormat
     [,1] [,2] [,3] [,4] [,5]
[1,]  1.0  0.5  0.5  0.5  0.5
[2,]  0.5  1.0  0.5  0.5  0.5
[3,]  0.5  0.5  1.0  0.5  0.5
[4,]  0.5  0.5  0.5  1.0  0.5
[5,]  0.5  0.5  0.5  0.5  1.0
```

```
> cholmat <- chol(cormat)
> cholmat
     [,1]      [,2]      [,3]      [,4]      [,5]
[1,]    1 0.5000000 0.5000000 0.5000000 0.5000000
[2,]    0 0.8660254 0.2886751 0.2886751 0.2886751
[3,]    0 0.0000000 0.8164966 0.2041241 0.2041241
[4,]    0 0.0000000 0.0000000 0.7905694 0.1581139
[5,]    0 0.0000000 0.0000000 0.0000000 0.7745967
```

```
> y <- x%*%beta0 +err
> y
            [,1]
 [1,] -10.5664923
 [2,]  -9.4427278
 [3,]   0.9429541
 [4,]  -0.3669282
 [5,]  -1.3057390
 [6,]  -1.9517771
 [7,]   2.1668896
 [8,]  15.0611422
 [9,]   7.4564457
[10,]   4.8290760
```

```
> x <- matrix(rnorm(5*n,0,1), ncol=5)%*%cholmat
> x
             [,1]        [,2]        [,3]        [,4]        [,5]
  [1,]  0.311731381 -0.66369847  0.041140041 -0.374405192  2.144010242
  [2,]  0.849829114  0.84263280  0.829036642  1.204984744  0.540948357
  [3,]  0.705533125  0.93532672  0.935272515  1.015216356  0.052834750
  [4,]  1.699928437  0.85942428  2.239938823  0.390866198  0.662927357
  [5,] -1.345570971 -1.63584251 -0.840874248 -0.505340414 -0.930967257
  [6,] -0.569813389 -1.03435069 -0.351191572  0.304097144 -0.871319368
  [7,]  0.059551728  0.24742778 -0.027658864 -0.643894229  0.230041389
  [8,] -1.434889351 -1.44693600 -1.909771693 -0.832888176 -2.588525231
  [9,]  0.196293496 -0.18122199  0.159855903  1.504167616 -0.595775647
 [10,]  0.026733743  0.40508637  0.079662870 -0.581299398  1.206466181
```

```
> err <- rnorm(n,0,sigma)
> err
  [1] -5.73148612 -9.45377843  0.07819296 -1.56999002 -3.75340631 -5.62331113
  [7]  3.89048949  9.05924727  3.32999225  9.74664776  2.40059912  0.53452419
 [13] -2.86327442 -9.02259765 -8.55830169 -1.56370874  8.75891227 10.27507307
 [19]  0.70874348 -6.67003085  5.03404196 -9.08210368  0.57132192 -4.82008131
 [25] -7.13158598  6.28282937 -1.00380282  4.15027662 -5.44269347  2.50337168
 [31]  3.25209421 -8.76066207 -1.26226922  3.46470362  0.11917389 -3.21672755
 [37]  3.37220787  4.54463739 -5.22655164  4.87602387 -1.49605270 -2.72216487
 [43] -7.84671146  3.27210662  2.25748499  3.87853891 -4.29768194 13.39067305
 [49]  1.21923362  8.91694507  7.37383797 -1.95223787 -2.09684440 -5.90180535
 [55]  2.28498271  0.70579049  3.65844136 -5.38942812  0.33404005 -2.42600651
 [61]  8.33480648  3.70135141 -9.83147374  7.08190348  6.46285127 -6.81173069
```

2.

2(a)

beta-hat = 88.27233, beta-tilde = 77.1113

- beta-hat：用 1.所得出的 matrix x 和 y、使用最小平方法(OLS)得到 beta-hat
- beta-tilde：把 x 和 y 做 rescale(x 做標準化、y 做中心化)，接著再使用最小平方法(OLS)做計算

實作方式：

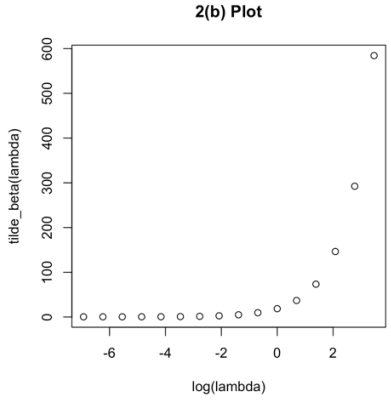
- 用一個 for 迴圈計算 yi-(β1xi1+β2xi2+…)，平方後用一個變數累加

2(b)

實作方式：

- 用 scale 過的 x,y 和題目指定的 lambda 得到 beta-tilde
- 為了跟 2(c)比較，把 lambda 取 log，x 軸為 log(lambda)

作圖如下：




```
+   square = (y[i] - x[i,1]*2 + x[i,2]*2 - x[i,3]*0.5
+   y_xbeta = y_xbeta + square
+ }
> y_xbeta = y_xbeta/(2*n)
> lambda_tilde_beta[2] = y_xbeta + 2*beta_square
> lambda_tilde_beta
              [,1]       [,2]
 [1,] 3.200000e+01 584.3855565
 [2,] 3.694136e+01 292.3855565
 [3,] 8.000000e+00 146.3855565
 [4,] 4.000000e+00  73.3855565
 [5,] 2.000000e+00  36.8855565
 [6,] 1.000000e+00  18.6355565
 [7,] 5.000000e-01   9.5105565
 [8,] 2.500000e-01   4.9480565
 [9,] 1.250000e-01   2.6668065
[10,] 6.250000e-02   1.5261815
[11,] 3.125000e-02   0.9558690
[12,] 1.562500e-02   0.6707127
[13,] 7.812500e-03   0.5281346
[14,] 3.906250e-03   0.4568455
[15,] 1.953125e-03   0.4212010
[16,] 9.765625e-04   0.4033787
```

估計 beta_hat(2)，使用與上面類似的方法，惟 x 和 y 是沒有 scale 過的，

beta_hat(2) = 36.94136

```
> lambda_hat_beta
     [,1]     [,2]
[1,]    2 36.94136
```

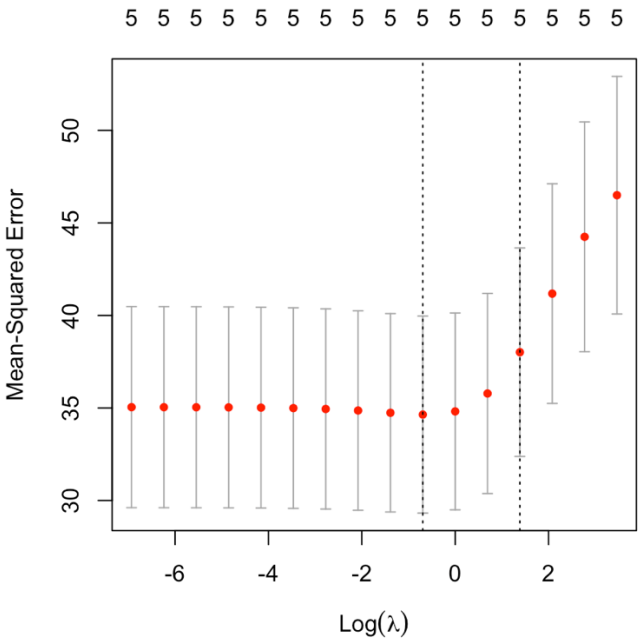
2(c)

同樣給定上題 lambda 的數列，用 lambda 參數直接給定值，下圖為使用 glmnet 後的結果與繪製的圖：

```
Call:  cv.glmnet(x = scale_x, y = scale_y, lambda = c(2^5, 2^4, 2^3,      2^2, 2^1, 2^0, 2^-1, 2^-2, 2^-3,
 2^-4, 2^-5, 2^-6, 2^-7,      2^-8, 2^-9, 2^-10), type.measure = "mse", nfolds = 5, family = "gaussian",
    alpha = 0)

Measure: Mean-Squared Error

    Lambda Measure    SE Nonzero
min    0.5   34.64 5.325       5
1se    4.0   38.01 5.635       5
```



下面為 lambda = 0.5、lambda = 4 分別的 beta 係數：

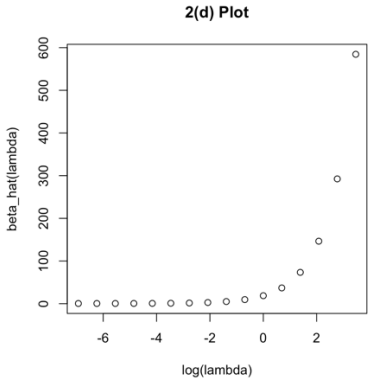
```
> # Use lambda.min                          > # Use lambda.1se
> ridge1 = glmnet(x,y,family = "gaussian", > ridge2 = glmnet(x,y,family = "gaussian",
> ridge1$beta   #係數                       > ridge2$beta   #係數
5 x 1 sparse Matrix of class "dgCMatrix"   5 x 1 sparse Matrix of class "dgCMatrix"
           s0                                          s0
V1  3.9423877                              V1  2.0415370
V2 -1.2297512                              V2 -0.7764445
V3 -0.3017515                              V3 -0.1037636
V4  1.0339691                              V4  0.4691393
V5 -3.5475998                              V5 -1.9545047
```

比較：

與 2(b)相比，最佳的 lambda 不同、但圖的趨勢大致相同，主要是因為用 glmnet 套件的 beta 值是根據 mse 的大小去尋找最適合的 beta，然而 2(b)的 beta 是題目給訂的，故無法得到最佳解。此外，2(b) y 軸的值也因為 beta 沒有選到最佳解而比 2(c)的值大許多。

2(d)

最佳的 lambda 為 2^(-10)，有最小的 beta_hat，結果如下圖：




```
                [,1]        [,2]
 [1,] 3.200000e+01 584.4413616
 [2,] 1.600000e+01 292.4413616
 [3,] 8.000000e+00 146.4413616
 [4,] 4.000000e+00  73.4413616
 [5,] 2.000000e+00  36.9413616
 [6,] 1.000000e+00  18.6913616
 [7,] 5.000000e-01   9.5663616
 [8,] 2.500000e-01   5.0038616
 [9,] 1.250000e-01   2.7226116
[10,] 6.250000e-02   1.5819866
[11,] 3.125000e-02   1.0116741
[12,] 1.562500e-02   0.7265179
[13,] 7.812500e-03   0.5839398
[14,] 3.906250e-03   0.5126507
[15,] 1.953125e-03   0.4770062
[16,] 9.765625e-04   0.4591839
```

實作方式：

- 與 2(b)類似，計算各個 lambda 分別的 beta_hat
- 選擇最小的 beta_hat
- 故最佳的 lambda 為 2^(-10)

附錄：R 語言程式碼

```
# HW3-106070038
## 1.
library(glmnet)
set.seed(36)
n<-100
sigma <- 5
beta0 <- c(2,-2,0.5,1,-3)
cormat <- diag(1,nrow=5,ncol=5)
cormat[cormat==0] <- 0.5
cholmat <- chol(cormat)
x <- matrix(rnorm(5*n,0,1), ncol=5)%*%cholmat
err <- rnorm(n,0,sigma)
y <- x%*%beta0 +err

## 2.
## (2a)
beta_hat = 0 #init
for (i in 100) {
      y_x_beta = y[i] - x[i,1]*2 + x[i,2]*2 - x[i,3]*0.5 - x[i,4]*1 + x[i,5]*3
      square = y_x_beta^2
      beta_hat = beta_hat + square
}
beta_hat #print

### scale x and y
scale_x = scale(x)
scale_y = y - mean(y)

beta_tilde = 0 #init
for (i in 100) {
   y_x_beta = scale_y[i] - scale_x[i,1]*2 + scale_x[i,2]*2 - scale_x[i,3]*0.5 -
scale_x[i,4]*1 + scale_x[i,5]*3
```

```
   square = y_x_beta^2
   beta_tilde = beta_tilde + square
}
beta_tilde #print

## (2b)
lambda_tilde_beta <- matrix( nrow = 16, ncol = 2)
lambda <- c(2^5,2^4,2^3,2^2,2^1,2^0,2^-1,2^-2,2^-3,2^-4,2^-5,2^-6,2^-7,2^-8,2^-
9,2^-10)
lambda
lambda_tilde_beta[,1]<- c(2^5,2^4,2^3,2^2,2^1,2^0,2^-1,2^-2,2^-3,2^-4,2^-5,2^-
6,2^-7,2^-8,2^-9,2^-10)
#lambda_tilde_beta
beta_square = 2^2+(-2)^2+0.5^2+1^2+(-3)^2
y_xbeta = 0
for (i in 100) {
   square = (scale_y[i] - scale_x[i,1]*2 + scale_x[i,2]*2 - scale_x[i,3]*0.5 -
   scale_x[i,4]*1 + scale_x[i,5]*3)^2
   y_xbeta = y_xbeta + square
}
buf = 2*n
y_xbeta = y_xbeta/buf
for (j in c(1:16)) {
   lambda_tilde_beta[j,2] = y_xbeta + lambda[j]*beta_square
}
lambda_tilde_beta
graphics.off()
plot(log(lambda_tilde_beta[,1]), lambda_tilde_beta[,2], xlab="log(lambda)",
        ylab="tilde_beta(lambda)",main="2(b) Plot")

## beta_hat_2
lambda = 2
lambda_hat_beta <- matrix( nrow = 1, ncol = 2)
```

```
lambda_hat_beta[1]<- 2
beta_square = 2^2+(-2)^2+0.5^2+1^2+(-3)^2
y_xbeta = 0
for (i in 100) {
   square = (y[i] - x[i,1]*2 + x[i,2]*2 - x[i,3]*0.5 - x[i,4]*1 + x[i,5]*3)^2
   y_xbeta = y_xbeta + square
}
y_xbeta = y_xbeta/(2*n)
lambda_hat_beta[2] = y_xbeta + lambda*beta_square
lambda_hat_beta

## (2c)
CVRIDGE = cv.glmnet(scale_x,scale_y,family = "gaussian",type.measure
='mse',nfold = 5,alpha = 0,lambda =c(2^5,2^4,2^3,2^2,2^1,2^0,2^-1,2^-2,2^-3,2^-
4,2^-5,2^-6,2^-7,2^-8,2^-9,2^-10))
CVRIDGE
plot(CVRIDGE)
# Use lambda.min
ridge1 = glmnet(x,y,family = "gaussian",alpha = 0,lambda = CVRIDGE$lambda.min)
# lambda 代 lambda.min
ridge1$beta     #係數

# Use lambda.1se
ridge2 = glmnet(x,y,family = "gaussian",alpha = 0,lambda = CVRIDGE$lambda.1se)
# lambda 代 lambda.1se
ridge2$beta     #係數

## (2d)
lambda_hat_beta <- matrix( nrow = 16, ncol = 2)
lambda <- c(2^5,2^4,2^3,2^2,2^1,2^0,2^-1,2^-2,2^-3,2^-4,2^-5,2^-6,2^-7,2^-8,2^-
9,2^-10)
lambda_hat_beta[,1]<- c(2^5,2^4,2^3,2^2,2^1,2^0,2^-1,2^-2,2^-3,2^-4,2^-5,2^-6,2^-
7,2^-8,2^-9,2^-10)
```

```
beta_square = 2^2+(-2)^2+0.5^2+1^2+(-3)^2
y_xbeta = 0
for (i in 100) {
   square = (y[i] - x[i,1]*2 + x[i,2]*2 - x[i,3]*0.5 - x[i,4]*1 + x[i,5]*3)^2
   y_xbeta = y_xbeta + square
}
y_xbeta = y_xbeta/(2*n)
for (j in c(1:16)) {
   lambda_hat_beta[j,2] = y_xbeta + lambda[j]*beta_square
}
lambda_hat_beta
graphics.off()
plot(log(lambda_hat_beta[,1]), lambda_hat_beta[,2], xlab="log(lambda)",
        ylab="beta_hat(lambda)",main="2(d) Plot")
```